The parameterized complexity of counting problems

Jörg Flum* Martin Grohe†

January 22, 2004

Abstract

We develop a parameterized complexity theory for counting problems. As the basis of this theory, we introduce a hierarchy of parameterized counting complexity classes #W$[t]$, for $t \geq 1$, that corresponds to Downey and Fellows's W-hierarchy [13] and show that a few central W-completeness results for decision problems translate to #W-completeness results for the corresponding counting problems.

Counting complexity gets interesting with problems whose decision version is tractable, but whose counting version is hard. Our main result states that counting cycles and paths of length k in both directed and undirected graphs, parameterized by k, is #W$[1]$-complete. This makes it highly unlikely that any of these problems is fixed-parameter tractable, even though their decision versions are fixed-parameter tractable. More explicitly, our result shows that most likely there is no $f(k) \cdot n^c$-algorithm for counting cycles or paths of length k in a graph of size n for any computable function $f : \mathbb{N} \to \mathbb{N}$ and constant c, even though there is a $2^{O(k)} \cdot n^{2.376}$ algorithm for finding a cycle or path of length k [2].

1 Introduction

Counting problems have been the source for some of the deepest and most fascinating results in computational complexity theory, ranging from Valiant's fundamental result [29] that counting perfect matchings of bipartite graphs is #P-complete over Toda's theorem [28] that the class $\mathrm{P}^{\#\mathrm{P}}$ contains the polynomial hierarchy to Jerrum, Sinclair, and Vigoda's [20] fully polynomial randomised approximation scheme for computing the number of perfect matchings of a bipartite graph. In this paper, we develop a basic parameterized complexity theory for counting problems.

Parameterized complexity theory provides a framework for a fine-grain complexity analysis of algorithmic problems that are intractable in general. In recent years, ideas from parameterized complexity theory have found their way into various areas of computer science, such as database theory [19, 24], artificial intelligence [18], and computational biology [6, 27]. Central to the theory is the notion of *fixed-parameter tractability*, which relaxes the classical notion of tractability, polynomial time computability, by admitting algorithms whose running time is exponential, but only in terms of some *parameter* of the problem instance that can be expected to be small in the typical applications. A good example is the evaluation of database queries: Usually, the size k of the query to be evaluated is very small compared to the size n of the database. An algorithm evaluating the query in time $O(2^k \cdot n)$ may therefore be acceptable, even quite good. On the other hand, an $\Omega(n^k)$ evaluation algorithm can usually not be considered feasible. Fixed-parameter tractability is based on this distinction: A parameterized problem is *fixed-parameter tractable* if there is a computable function f and a constant c such that the problem can be solved in time $f(k) \cdot n^c$, where n is the input size and k the parameter value.

A standard example of a fixed-parameter tractable problem is the vertex cover problem parameterized by the size k of the vertex cover. It is quite easy to see that a vertex cover of size k of a graph of size n can be computed in time $O(2^k \cdot n)$ by a simple search tree algorithm based on the fact that at least one of the two endpoints of each edge must be contained in a vertex cover. (As a matter of fact, such an algorithm computes all minimum vertex covers of size at most k). A standard example of a problem that does not

*Institut für Mathematische Logik, Albert-Ludwigs-Universität Freiburg, Eckerstr. 1, 79104 Freiburg, Germany. Email: `Joerg.Flum@math.uni-freiburg.de`

†Institur für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany. Email: `grohe@informatik.hu-berlin.de`

seem to be fixed-parameter tractable is the clique problem, parameterized by the size of the clique. Indeed, all known algorithms for deciding whether a graph of size n has a clique of size k have a running time of $n^{\Omega(k)}$.

To give evidence that parameterized problems such as the clique problem are not fixed parameter tractable, a theory of *parameterized intractability* has been developed (see [11, 12, 13]). It resulted in a rather unwieldy variety of parameterized complexity classes. The most important of these classes are the classes W$[t]$, for $t \geq 1$, forming the so-called *W-hierarchy*. It is believed that W$[1]$ strictly contains the class FPT of all fixed-parameter tractable problems and that the W-hierarchy is strict. Many natural parameterized problems fall into one of the classes of the W-hierarchy. For example, the parameterized clique problem is complete for the class W$[1]$ and the parameterized dominating set problem is complete for the class W$[2]$ (under suitable parameterized reductions).

So far, the parameterized complexity of counting problems has not been studied very systematically. A few tractability results are known: First of all, some fixed-parameter tractable decision problems have algorithms that can easily be adapted to the corresponding counting problems. An example is the vertex cover problem; since all minimum vertex covers of size at most k of a graph of size n can be computed in time $O(2^k \cdot n)$, a simple application of the inclusion-exclusion principle yields a fixed-parameter tractable counting algorithm for the vertex covers of size k. Similar counting algorithms are possible for other problems that have a fixed-parameter tractable algorithm based on the *method of bounded search tree* (see [13]). More interesting are results of Arnborg, Lagergren, and Seese [4], Courcelle, Makowsky, and Rotics [10], and Makowsky [22] stating that counting problems definable in monadic second-order logic (in various ways) are fixed-parameter tractable when parameterized by the tree-width of the input graph. For example, Arnborg et al's result implies that counting the Hamiltonian cycles of a graph is fixed-parameter tractable when parameterized by the tree-width of the graph, and Makowsky's result implies that evaluating the Tutte polynomial is fixed-parameter tractable when parameterized by the tree-width of the graph. Courcelle et al. [10] also proved similar results for graphs of bounded clique-width. Frick [16] showed that counting problems definable in first-order logic are fixed-parameter tractable on locally tree-decomposable graphs. For example, this implies that counting dominating sets of a planar graph is fixed-parameter tractable when parameterized by the size of the dominating sets.

We focus on the *intractability* of parameterized counting problems. We define classes #W$[t]$, for $t \geq 1$, of parameterized counting problems that correspond to the classes of the W-hierarchy. Our first results show that a few central completeness results for the classes W$[1]$ and W$[2]$ translate to corresponding completeness results for the first two levels #W$[1]$ and #W$[2]$ of the #W-hierarchy. For example, we show that counting cliques of size k is #W$[1]$-complete and counting dominating sets of size k is #W$[2]$-complete (both under *parsimonious parameterized reductions*). We then characterise the class #W$[1]$ as the class of all counting problems that can be described in terms of numbers of accepting computations of certain nondeterministic programs. To give further evidence that the class #W$[1]$ strictly contains the class of fixed-parameter tractable counting problems, we show that if this was not the case there would be a $2^{o(n)}$-algorithm counting the satisfying assignments of a 3-CNF-formula with n variables. This is the counting version of a result due to Abrahamson, Downey, and Fellows [1]. While these results are necessary to lay a solid foundation for the theory and not always easy to prove, by and large they do not give us remarkable new insights. The theory gets interesting with those counting problems that are harder than their decision versions.

Our main result states that *counting cycles and paths of length k in both directed and undirected graphs, parameterized by k, is* #W$[1]$*-complete under parameterized Turing reductions*. It is an immediate consequence of a theorem of Plehn and Voigt [26] that the decision versions of these problems are fixed-parameter tractable (but of course not in polynomial time, because if they were the Hamiltonian path/cycle problem would also be). Alon, Yuster, and Zwick's [2] *color coding* technique provides algorithms for finding a path of length k in time $O(k! \cdot m)$ in a graph with m edges and for finding a cycle of length k in time $O(2^{O(k)} \cdot n^{\omega})$ in a graph with n vertices, where $\omega < 2.376$ is the exponent of matrix multiplication. The hardness of the cycle counting problem in undirected graphs may be surprising in view of another algorithm due to Alon, Yuster, and Zwick [3] showing that cycles *up to length 7* in an undirected graph can be counted in time $O(n^{\omega})$. Our result implies that it is very unlikely that there is such an algorithm for counting cycles of arbitrary fixed length k.

The paper is organised as follows: After giving the necessary preliminaries in Section 2, in Section 3 we discuss fixed-parameter tractable counting problems. This section has the character of a short survey; apart from a few observations it contains no new results. In Section 4, we introduce the #W-hierarchy and establish the basic completeness results. The hardness of counting cycles and paths is established in Section 5. Definitions of all parameterized problems considered in this paper can be found in Appendix A.

We would like to point out that some of the results in Section 4 have independently been obtained by others in two recent papers: McCartin [23] proves the #W[1]-completeness of clique and the #W[2]-completeness of dominating set. (Our proofs of these results are quite different from hers.) Furthermore, she shows that a number of further completeness results for parameterized decision problems translate to the corresponding counting problems. Arvind and Raman [5] also obtain the #W[1]-completeness of clique. Their main result is that the number of cycles or paths of length k can be approximated by a randomised fixed-parameter tractable algorithm. Indeed, they prove this not only for cycles and paths, but for arbitrary graphs of bounded tree-width. These results nicely complement our main result that exactly counting paths and cycles is hard.

2 Preliminaries

2.1 Parameterized Complexity Theory A *parameterized problem* is a set $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. If $(x,k) \in \Sigma^* \times \mathbb{N}$ is an instance of a parameterized problem, we refer to x as the *input* and to k as the *parameter*.

Definition 1 A parameterized problem $P \subseteq \Sigma^* \times \mathbb{N}$ is *fixed-parameter tractable* if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, a constant $c \in \mathbb{N}$, and an algorithm that, given a pair $(x,k) \in \Sigma^* \times \mathbb{N}$, decides if $(x,k) \in P$ in at most $f(k) \cdot |x|^c$ steps.

We usually use k to denote the parameter and $n = |x|$ to denote the size of the input.

To illustrate our notation, let us give one example of a parameterized problem, the *parameterized vertex cover problem*, which is well-known to be fixed-parameter tractable:

p-VC	
Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ has a vertex cover of size k.

From now on, we will only give brief definitions of the parameterized problems we consider in the main text, for exact definitions we refer the reader to Appendix A.

To define the classes of the W-hierarchy, we need a few notions from propositional logic. Formulas of propositional logic are built up from *propositional variables* $X_1, X_2, \dots$ by taking conjunctions, disjunctions, and negations. The negation of a formula θ is denoted by $\neg\theta$. We distinguish between *small conjunctions,* denoted by $\wedge$, which are just conjunctions of two formulas, and *big conjunctions,* denoted by $\bigwedge$, which are conjunctions of arbitrary finite sets of formulas. Analogously, we distinguish between *small disjunctions,* denoted by $\vee$, and *big disjunctions,* denoted by $\bigvee$.

A formula is *small* if it only contains small conjunctions and small disjunctions. We define $\Gamma_0 = \Delta_0$ to be the class of all small formulas. For $t \geq 1$, we define Γ_t to be the class of all big conjunctions of formulas in Δ_{t-1}, and we define Δ_t to be the class of all big disjunctions of formulas in Γ_{t-1}.

The *depth* of a propositional formula θ is the maximum number of nested conjunctions or disjunctions in θ. Note that the definitions of Γ_t and Δ_t are purely syntactical; every formula in Γ_t or Δ_t is equivalent to a formula in Γ_0. But the translation from a formula in Γ_t to an equivalent formula in Γ_0 usually increases the depth of a formula. For all $t, d \geq 0$ we let $\Gamma_{t,d}$ denote the class of all formulas in Γ_t whose small subformulas have depth at most d (equivalently, we may say that the whole formula has depth at most $d + t$). We define $\Delta_{t,d}$ analogously.

Let CNF denote the class of all propositional formulas in conjunctive normal form, that is, conjunctions of disjunctions of literals; if we ignore arbitrarily nested negations then CNF is just $\Gamma_{2,0}$. A formula is in

d *conjunctive normal form* if it is a conjunction of disjunctions of at most d literals; the class of all such formulas is denoted by d-CNF.

The *weight* of a truth value assignment to the variables of a propositional formula is the number of variables set to TRUE by the assignment. For any class Θ of propositional formulas, the *weighted satisfiability problem for* Θ, denoted by $\text{WSAT}(\Theta)$, is the problem of deciding whether a formula in Θ has a satisfying assignment of weight k, parameterized by k. We are now ready to define the W-hierarchy:

Definition 2 For $t \geq 1$, W$[t]$ is the class of all parameterized problems that can be reduced to $\text{WSAT}(\Gamma_{t,d})$ for some $d \geq 0$ by a parameterized many-one reduction.

We omit the definition of parameterized many-one reductions here and refer the reader to [13] for this definition and further background on parameterized complexity theory.

2.2 Relational Structures A *vocabulary* is a finite set of relation symbols. Associated with every relation symbol is a natural number, its *arity*. The arity of a vocabulary is the maximum of the arities of the relation symbols it contains. In the following, τ always denotes a vocabulary.

A τ-*structure* $\mathcal{A}$ consists of a non-empty set A, called the *universe* of $\mathcal{A}$, and a relation $R^{\mathcal{A}} \subseteq A^r$ for each r-ary relation symbol $R \in \tau$. For example, we view a *directed graph* as a structure $\mathcal{G} = (G, E^{\mathcal{G}})$ whose vocabulary consists of one binary relation symbol E. $\mathcal{G} = (G, E^{\mathcal{G}})$ is an *(undirected) graph* if $E^{\mathcal{G}}$ is symmetric. For graphs, we often write $\{a, b\} \in E^{\mathcal{G}}$ instead of $(a, b) \in E^{\mathcal{G}}$. In this paper, we only consider structures whose universe is finite. We distinguish between the size of the universe A of a τ-structure $\mathcal{A}$, which we denote by $|A|$, and the *size* of $\mathcal{A}$, which is defined to be

$$||\mathcal{A}|| = |\tau| + |A| + \sum_{R \in \tau} |R^{\mathcal{A}}| \cdot \text{arity}(R).$$

An *expansion* of a τ-structure $\mathcal{A}$ to a vocabulary $\tau' \supseteq \tau$ is a τ'-structure $\mathcal{A}'$ with $A' = A$ and $R^{\mathcal{A}'} = R^{\mathcal{A}}$ for all $R \in \tau$.

A *substructure* of $\mathcal{A}$ is a structure $\mathcal{B}$ with $B \subseteq A$ and $R^{\mathcal{B}} \subseteq R^{\mathcal{A}}$ for all $R \in \tau$.[1] A *homomorphism* from a τ-structure $\mathcal{A}$ to a τ-structure $\mathcal{B}$ is a mapping $h : A \to B$ where for all $R \in \tau$, say, of arity r, and all tuples $(a_1, \ldots, a_r) \in R^{\mathcal{A}}$ we have $\big(h(a_1), \ldots, h(a_r)\big) \in R^{\mathcal{B}}$. An *embedding* is a homomorphism that is one-to-one.

The *homomorphism problem* asks whether there is a homomorphism from a given structure $\mathcal{A}$ to a given structure $\mathcal{B}$. We parameterize this problem by the size of $\mathcal{A}$ and denote the resulting *parameterized homomorphism problem* by p-HOM. We will also consider the *parameterized embedding problem*, denoted by p-EMB, and the *parameterized substructure problem* (Does structure $\mathcal{B}$ have a substructure isomorphic to $\mathcal{A}$?), denoted by p-SUB. Of course when considered as decision problems, p-EMB and p-SUB are equivalent, but as counting problems they are slightly different. All three decision problems are complete for the class W$[1]$ under parameterized many-one reductions [13].

2.3 Logic and Descriptive Complexity Let us remark that the following notions are not needed for understanding our results on the hardness of counting cycles and paths or their proofs.

The formulas of *first-order logic* are built up from *atomic formulas* using the usual Boolean connectives and existential and universal quantification over the elements of the universe of a structure. Remember that an *atomic formula*, or *atom*, is a formula of the form $x = y$ or $Rx_1 \ldots x_r$, where R is an r-ary relation symbol and $x, y, x_1, \ldots, x_r$ are variables. A *literal* is either an atom or a negated atom. The *vocabulary* of a formula φ is the set of all relation symbols occurring in φ. A *free variable* of a formula φ is a variable that is not bound by any existential or universal quantifier of φ.

If $\mathcal{A}$ is a τ-structure, $a_1, \ldots, a_n$ are elements of the universe A of $\mathcal{A}$, and $\varphi(x_1, \ldots, x_n)$ is a formula whose vocabulary is a subset of τ and whose free variables are $x_1, \ldots, x_n$, then we write $\mathcal{A} \models$

[1]Note that in logic, substructures are usually required to satisfy the stronger condition $R^{\mathcal{B}} = R^{\mathcal{A}} \cap A^r$, where r is the arity of R. Our notion of substructure is the direct generalisation of the standard graph theoretic notion of subgraph. Since we are mainly dealing with graphs, this seems appropriate. A similar remark applies to our notion of embedding.

$\varphi(a_1,\ldots,a_n)$ to denote that $\mathcal{A}$ satisfies φ if the variables $x_1,\ldots,x_n$ are interpreted by $a_1,\ldots,a_n$, respectively. We let

$$\varphi(\mathcal{A}) := \big\{(a_1,\ldots,a_n)\in A^n \;\big|\; \mathcal{A}\models\varphi(a_1,\ldots,a_n)\big\}.$$

To get a uniform notation, we let A^0 be a one-point space and identify $\emptyset$ with FALSE and A^0 with TRUE. Then for a sentence φ (i.e. a formula without free variables), we have $\mathcal{A}\models\varphi \iff \varphi(\mathcal{A}) = \text{TRUE}$. Furthermore, if the vocabulary of the formula φ is not contained in the vocabulary of $\mathcal{A}$ then we let $\varphi(\mathcal{A}) = \emptyset$.

For every class Φ of formulas, we let $\Phi[\tau]$ be the class of all $\varphi\in\Phi$ whose vocabulary is contained in τ. We let both Σ_0 and Π_0 be the class of all quantifier free first-order formulas (although we usually use Π_0 to denote this class). For $t\geq 1$, we let Σ_t be the class of all first-order formulas of the form $\exists x_1\ldots\exists x_k\;\psi$, where $k\in\mathbb{N}$ and $\psi\in\Pi_{t-1}$. Analogously, we let Π_t be the class of all first-order formulas of the form $\forall x_1\ldots\forall x_k\;\psi$, where $k\in\mathbb{N}$ and $\psi\in\Sigma_{t-1}$.

We have to define two additional hierarchies $(\Sigma_{t,u})_{t\geq 1}$ and $(\Pi_{t,u})_{t\geq 1}$ for every fixed $u\geq 1$. Again we let $\Sigma_{0,u} = \Pi_{0,u} = \Pi_0$. We let $\Pi_{1,u}$ be the class of all first-order formulas of the form $\forall x_1\ldots\forall x_k\;\psi$, where $k\leq u$ and $\psi\in\Pi_0$. For $t\geq 2$, we let $\Pi_{t,u}$ be the class of all first-order formulas of the form $\forall x_1\ldots\forall x_{k_1}\;\exists y_1\ldots\exists y_{k_2}\;\psi$, where $k_1,k_2\leq u$ and $\psi\in\Pi_{t-2,u}$. For $t\geq 1$, we let $\Sigma_{t,u}$ be the class of all first-order formulas of the form $\exists x_1\ldots\exists x_k\;\psi$, where $k\in\mathbb{N}$ and $\psi\in\Pi_{t-1,u}$. Note the asymmetry in the definitions of $\Pi_{t,u}$ and $\Sigma_{t,u}$ — the length of the first quantifier block in a $\Sigma_{t,u}$-formula is not restricted.

Definability of Parameterized Problems I: Model-Checking Problems

We can use logic to define certain generic families of parameterized problems. For a class Φ of formulas, the *model-checking problem* for Φ is the problem of deciding whether for a given structure $\mathcal{A}$ and a given formula $\varphi\in\Phi$ we have $\varphi(\mathcal{A})\neq\emptyset$. We parameterize this problem by the length of the formula φ and obtain the *parameterized model-checking problem* p-MC(Φ).

Many parameterized problems can be naturally translated into model-checking problems. For example, the parameterized clique problem is essentially the same as the parameterized model-checking problem for the class

$$\Phi_{\text{CLIQUE}} = \Big\{\bigwedge_{1\leq i<j\leq k}(Ex_ix_j\wedge x_i\neq x_j) \;\Big|\; k\geq 1\Big\}.$$

Model checking problems provide another basis for the W-hierarchy: For every $t\geq 1$, W$[t]$ is the class of all problems that are reducible to p-MC$(\Sigma_{t,1}[\tau])$ for some vocabulary τ by a parameterized many-one reduction [14, 15]. Observe, furthermore, that for all $t\geq 1$ the problems p-MC$(\Sigma_{t,1})$ and p-MC$(\Pi_{t-1,1})$ are easily reducible to each other, because for every formula

$$\varphi(x_1,\ldots,x_k) = \exists y_1\ldots\exists y_l\;\psi(x_1,\ldots,x_k,y_1,\ldots,y_l)$$

and every structure $\mathcal{A}$ we have $\varphi(\mathcal{A})\neq\emptyset$ if, and only if, $\psi(\mathcal{A})\neq\emptyset$. This explains why the hierarchies $(\Sigma_{t,u})_{t\geq 1}$ and $(\Pi_{t,u})_{t\geq 1}$ are defined asymmetrically.

Definability of Parameterized Problems II: Fagin-definability

There is a second way of defining parameterized problems that has been dubbed *Fagin-definability* in [15]. Let φ be a sentence of vocabulary $\tau\cup\{X\}$, where X is a relation symbol not contained in τ. We view X as a *relation variable*; to illustrate this we usually write $\varphi(X)$ instead of just φ. Let r be the arity of X. For a τ-structure $\mathcal{A}$ we let

$$\varphi(\mathcal{A}) = \big\{R\subseteq A^r \;\big|\; (\mathcal{A},R)\models\varphi\big\},$$

where $(\mathcal{A},R)$ denotes the $\tau\cup\{X\}$-expansion of $\mathcal{A}$ with $X^{(\mathcal{A},R)} = R$. For example, let X be unary and

$$\varphi_{\text{VC}}(X) = \forall y\forall z\big(Eyz\rightarrow(Xy\vee Xz)\big).$$

Then for a graph $\mathcal{G}$, $\varphi_{\text{VC}}(\mathcal{G})$ is the set of all vertex covers of $\mathcal{G}$.

With each formula $\varphi(X)$ we associate a parameterized problem p-FD$(\varphi(X))$ which asks whether for a given structure $\mathcal{A}$, the set $\varphi(\mathcal{A})$ contains a relation with k elements (where k is the parameter). We call p-FD$(\varphi(X))$ the problem *Fagin-defined* by $\varphi(X)$.

For example, p-FD$(\varphi_{\text{VC}}(X))$ is precisely the parameterized vertex cover problem.

Initialise $\mathcal{S} \subseteq \mathrm{Pow}(G)$ by $\mathcal{S} := \{\emptyset\}$
for all $\{a, b\} \in E^{\mathcal{G}}$ **do**
  for all $S \in \mathcal{S}$ **do**
    if $S \cap \{a, b\} = \emptyset$ **then**
      $\mathcal{S} := \mathcal{S} \setminus \{S\}$
      if $|S| < k$ **then** $\mathcal{S} := \mathcal{S} \cup \{S \cup \{a\}, S \cup \{b\}\}$.
output $\mathcal{S}$.

Algorithm 1

3 Tractable parameterized counting problems

A *parameterized counting problem* is simply a function $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$, for some alphabet Σ. Arguably, this definition includes problems that intuitively we would not call counting problems, but there is no harm in including them.

Definition 3 A parameterized counting problem $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$ is *fixed-parameter tractable*, or $F \in$ FPT, if there is an algorithm computing $F(x, k)$ in time $f(k) \cdot |x|^c$ for some computable function $f : \mathbb{N} \to \mathbb{N}$ and some constant $c \in \mathbb{N}$.

The standard example of a fixed-parameter tractable decision problem is the parameterized version of the vertex cover problem. As a first example, we observe that the corresponding counting problem is also fixed-parameter tractable:

Example 4 *The parameterized vertex cover counting problem,*

p-#VC

Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the number of vertex covers of $\mathcal{G}$ of size k.

is fixed-parameter tractable.

Proof: Essentially, Algorithm 1 is the standard procedure showing that the parameterized vertex cover problem is fixed-parameter tractable. It yields, given a graph $\mathcal{G} = (G, E^{\mathcal{G}})$ as input and $k \in \mathbb{N}$ as parameter, a set $\mathcal{S}$ of subsets of cardinality $\leq k$ of G in time $O(2^k \cdot \|\mathcal{G}\|)$ such that for the set $\mathrm{VC}_k(\mathcal{G})$ of vertex covers of $\mathcal{G}$ of cardinality k we have

$$\mathrm{VC}_k(\mathcal{G}) \quad = \quad \{X \subseteq G \mid |X| = k \text{ and } S \subseteq X \text{ for some } S \in \mathcal{S}\}.$$

Now, we can compute $|\mathrm{VC}_k(\mathcal{G})|$ by applying the inclusion-exclusion principle to the sets

$$\{X \subseteq G \mid |X| = k \text{ and } S \subseteq X\}$$

for $S \in \mathcal{S}$. □

We could now go through a list of known fixed-parameter tractable problems and check if the corresponding counting problems are also fixed-parameter tractable. Fortunately, this boring task can largely be avoided, because there are a few general principles underlying most fixed-parameter tractability results. They are formulated in the terminology of descriptive complexity theory.

(1) *Problems definable in monadic second-order logic are fixed-parameter tractable when parameterized by tree-width of the structure* (Courcelle [9]). This accounts for the fixed-parameter tractability of NP-complete problems such as 3-COLOURABILITY or HAMILTONICITY when parameterized by the tree-width of the input graph.

(2) *Parameterized problems that can be described as model-checking problems for first-order logic are fixed-parameter tractable on classes of structures of bounded local tree-width and classes of graphs with excluded minors* (Frick and Grohe [17], Flum and Grohe [15]). This implies that parameterized versions of problems such as dominating set, independent set, or subgraph isomorphism are fixed-parameter tractable on planar graphs or on graphs of bounded degree.

(3) *Parameterized problems that are Fagin-definable by a first-order formula $\varphi(X)$ where X does not occur in the scope of a negation symbol or existential quantifier are fixed-parameter tractable* (Cai and Chen [7], Flum and Grohe [15]). This accounts for the fixed-parameter tractability of the standard parameterization of minimisation problems in the classes MIN $F^+\Pi_1$ [21], for example, minimum vertex cover.

(4) *Parameterized problems that can be described as parameterized model checking problems for Σ_1-formulas of bounded tree-width are fixed-parameter tractable* (Flum and Grohe [15]). This implies, and is actually equivalent, to the results that the parameterized homomorphism problem and the parameterized embedding problem for relational structures of bounded tree-width are fixed-parameter tractable.

Let us consider the counting versions of these general "meta-theorems". It has already been proved by Arnborg, Lagergren, and Seese [4] that the counting version of (1) holds. Variants and extensions of this result have been proved by Courcelle, Makowsky, and Rotics [10] and Makowsky [22]. Frick proved that (most of) (2) also extends to counting problems [16]. We shall see below that (3) also extends to counting problems. (4) is more problematic. While the counting version of the parameterized homomorphism problem for structures of bounded tree-width is fixed-parameter tractable (actually in polynomial time), the general equivalence between homomorphism, embedding, and model-checking for Σ_1 breaks down for counting problems. This is the point where counting shows some genuinely new aspects, and in some sense, most of this paper is devoted to this phenomenon.

Let us turn to (3), the Fagin-definable problems, now: For a formula $\varphi(X)$, we let p-#FD$(\varphi(X))$ denote the natural counting version of the problem p-FD$(\varphi(X))$ Fagin-defined by $\varphi(X)$. Recalling the formula $\varphi_{\text{VC}}(X)$ that Fagin-defines the parameterized vertex cover problem, we see that the following proposition generalises Example 4.

Proposition 5 *Let $\varphi(X)$ be a first-order formula in which X does not occur in the scope of an existential quantifier or negation symbol. Then p-#FD$(\varphi(X)) \in$ FPT.*

Proof: Let X be of arity r. As for the vertex cover problem one obtains an FPT-algorithm (cf. [15]) that, given a structure $\mathcal{A}$ and $k \in \mathbb{N}$, yields a set $\mathcal{S}$ of subsets of cardinality $\leq k$ of A^r such that

$$\{X \subseteq A^r \mid |X| = k \text{ and } \mathcal{A} \models \varphi(X)\} \quad = \quad \{X \subseteq A^r \mid |X| = k \text{ and } S \subseteq X \text{ for some } S \in \mathcal{S}\}.$$

Again, an application of the inclusion-exclusion principle allows to compute the cardinality of the set on the right hand side. □

This implies that the counting versions of the standard parameterizations of all minimisation problems in the class MIN $F^+\Pi_1$ are fixed-parameter tractable.

As we have mentioned, the situation with item (4) in the list above is more complicated. The core problem for which counting remains tractable is the homomorphism problem for graphs of bounded tree-width. Again, the algorithm showing tractability can best be illustrated by an example.

Example 6 *The number of homomorphisms from a given coloured tree $\mathcal{T}$ to a given coloured graph $\mathcal{G}$ can be computed in polynomial time.*

This can be done by a simple dynamic programming algorithm. Starting from the leaves, for every vertex t of the tree we compute a table that stores, for all vertices v of the graph, the number $H(t, v)$ of homomorphisms h from $\mathcal{T}_t$, the induced coloured subtree rooted at t, to $\mathcal{G}$ with $h(t) = v$. Then the total number of homomorphism from $\mathcal{T}$ to $\mathcal{G}$ is $\sum_{v \in G} H(r, v)$, where r is the root of $\mathcal{T}$.

If t is a leaf, then $H(t, v) = 1$ if t and v have the same colour and $H(t, v) = 0$ otherwise. If t has children $t_1, \ldots, t_l$, then if t and v have the same colour we have

$$H(t, v) = \prod_{i=1}^{l} \sum_{\substack{w \in G \\ w \text{ adjacent to } v}} H(t_i, w).$$

If t and v have distinct colours, we have $H(t, v) = 0$.

The previous example can easily be generalised to structures of bounded tree-width. We just state the result and omit a definition of tree-width and the proof, which is a straightforward generalisation of the example:

Proposition 7 *Let $w \geq 1$. Then the following restriction of the homomorphism problem is in polynomial time:*

Input:	Structure $\mathcal{A}$ of tree-width at most w, structure $\mathcal{B}$.
Problem:	Count the homomorphisms from $\mathcal{A}$ to $\mathcal{B}$.

For a class Φ of formulas, we let p-#(Φ) denote the counting version of the model-checking problem p-MC(Φ) ("Given $\mathcal{A}$ and $\varphi \in \Phi$, compute $|\varphi(\mathcal{A})|$, parameterized by $|\varphi|$").

With every first-order formula φ we associate a graph $\mathcal{G}_\varphi$ as follows: The vertices of $\mathcal{G}_\varphi$ are the variables of φ, and there is an edge between two vertices if they occur together in an atomic subformula of φ. The *tree-width* of a formula φ is the tree-width of $\mathcal{G}_\varphi$. For a class Φ of formulas and $w \geq 1$, we let $\Phi[\text{tw } w]$ denote the class of all formulas in Φ of tree-width at most w. Recall that Π_0 denotes the class of all quantifier free formulas.

Proposition 8 *For every $w \geq 1$ we have p-#$(\Pi_0[\text{tw } w]) \in$ FPT.*

Proof: We can effectively transform every $\varphi(\bar{x}) \in \Pi_0[\text{tw } w]$ into an equivalent $\psi(\bar{x}) \in \Pi_0[\text{tw } w]$ in disjunctive normal form, $\psi(\bar{x}) = \psi_1(\bar{x}) \vee \ldots \vee \psi_r(\bar{x})$, where each $\psi_i(\bar{x})$ is a conjunction of literals and where $\psi_i(\bar{x}) \wedge \psi_j(\bar{x})$ is unsatisfiable for all i, j with $i \neq j$. Note that this transformation does not change the tree-width of the formula because the set of atomic subformulas remains unchanged. Then for every structure $\mathcal{A}$ we have

$$|\psi(\mathcal{A})| = |\psi_1(\mathcal{A})| + \ldots + |\psi_r(\mathcal{A})|.$$

Thus we can restrict our attention to formulas in $\Pi_0[\text{tw } w]$ that are conjunctions of literals. Since every literal of a formula whose underlying graph has tree-width at most w contains at most $w + 1$ variables, by standard techniques (cf. the proof of Theorem 14) the counting problem for such formulas can be reduced to the counting version of the homomorphism problem for structures of tree-width at most w, which is in polynomial time by Proposition 7. □

Remark 9 Note that although its core is a reduction to Proposition 7, the proof of the previous proposition does *not* yield a polynomial time algorithm. The reason is that the transformation of a formula to an equivalent formula in disjunctive normal form is not polynomial.

Indeed, it is easy to see that the unparameterized counting problem for quantifier free formulas of tree-width 0 is #P-complete.

Clearly, if p-MC(Φ) is fixed-parameter tractable then so is p-MC(Φ^*), where Φ^* is the closure of Φ under existential quantification. Thus in particular, for $w \geq 1$ the problem p-MC(Σ_1[tw w]) is fixed-parameter tractable. The situation is different for the counting problems: The formula

$$\varphi(x_1, \ldots, x_k) \quad := \quad \exists y \bigwedge_{i=1}^{k} (\neg Eyx_i \wedge \neg y = x_i)$$

is a Σ_1-formula of tree-width 1. For all graphs $\mathcal{G}$, the set $\varphi(\mathcal{G})$ is the set of all tuples $(a_1, \ldots, a_k)$ of vertices of $\mathcal{G}$ such that $\{a_1, \ldots, a_k\}$ is *not* a dominating set. Thus $\mathcal{G}$ has a dominating set of size at most k if, and only if, $|\varphi(\mathcal{G})| < n^k$, where n is the number of vertices of $\mathcal{G}$. Since the parameterized dominating set problem is complete for the class W[2] this implies:

Proposition 10 *If* W[2] $\neq$ FPT *then* p-#(Σ_1[tw w]) $\notin$ FPT.

4 Classes of intractable problems

Example 11 Valiant's [29] fundamental theorem states that counting the number of perfect matchings of a bipartite graph is #P-complete (whereas deciding whether a perfect matching exists is in P). We consider a *trivial parameterization* of the matching problem, which is obtained by adding a "dummy" parameter as follows:

Input:	Bipartite Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ has a perfect matching.

Clearly, this problem is in polynomial time and thus fixed-parameter tractable. On the other hand, its counting version ("Count the perfect matchings of $\mathcal{G}$.") cannot be fixed-parameter tractable unless P = #P. The reason for this is that the problem is already #P-complete for the fixed parameter value $k = 1$, but if it was fixed-parameter tractable it would be in polynomial time for any fixed parameter value.

Of course this example is quite artificial. We are more interested in the question of whether natural parameterized counting problems are fixed-parameter tractable. As examples of such natural problems we mention: p-#CLIQUE ("Count cliques of size k in a graph, where k is the parameter"), p-#DOMINATING SET ("Count dominating sets of size k"), p-#CYCLE ("Count cycles of size k"), or as a more natural parameterization of the matching problem, p-#MATCHING ("Count the matchings of size k in a bipartite graph"). An argument such as the one in Example 11 cannot be used to show that any of these problems is not fixed-parameter tractable, because for any fixed parameter value k the problems are in polynomial time.

Recall that the decision problems p-CLIQUE and p-DOMINATING SET are complete for the classes W[1] and W[2], respectively, so the counting problems cannot be fixed-parameter tractable unless W[1] = FPT (W[2] = FPT, respectively). We will define classes #W[t] of counting problems and show for a few central W[1]-complete and W[2]-complete problems that their counting versions are #W[1]-complete (#W[2]-complete, respectively). More interestingly, in the next section we shall prove that p-#CYCLE and a number of similar problems whose decision versions are fixed-parameter tractable are complete for #W[1].

Definition 12 Let $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$ and $G : \Pi^* \times \mathbb{N} \to \mathbb{N}$ be parameterized counting problems.

(1) A *parameterized parsimonious reduction* from F to G is an algorithm that computes for every instance (x, k) of F an instance (y, ℓ) of G in time $f(k) \cdot |x|^c$ such that $\ell \leq g(k)$ and

$$F(x, k) = G(y, \ell)$$

(for computable functions $f, g : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$).

We write $F \leq^{\text{fp}}_{\text{pars}} G$ to denote that there is a parameterized parsimonious reduction from F to G.

(2) A *parameterized T-reduction* from F to G is an algorithm with an oracle for G that solves any instance (x, k) of F in time $f(k) \cdot |x|^c$ in such a way that for all oracle queries the instances (y, ℓ) satisfy $\ell \leq g(k)$ (for computable functions $f, g : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$).

We write $F \leq^{\text{fp}}_{\text{T}} G$ to denote that there is a parameterized T-reduction from F to G.

Obviously, if $F \leq^{\text{fp}}_{\text{pars}} G$ then $F \leq^{\text{fp}}_{\text{T}} G$. An easy computation shows that if $G \in$ FPT and $F \leq^{\text{fp}}_{\text{T}} G$ then $F \in$ FPT.

For a class Θ of propositional formulas, we let #WSAT(Θ) be the counting version of the weighted satisfiability problem for Θ ("Count the weight k satisfying assignments for a formula $\theta \in \Theta$"). We define the counting analogue of the W-hierarchy in a straightforward way:

Definition 13 For $t \geq 1$, #W$[t]$ is the class of all parameterized counting problems that are fixed-parameter parsimonious reducible to #WSAT$(\Gamma_{t,d})$, for some $d \geq 0$.

The notation #W$[t]$ may be slightly misleading when compared with the notation #P of classical complexity theory (which is not #NP), but since there is no obvious #FPT, we think that it is appropriate. Note that we write FPT to denote both the class of fixed-parameter tractable decision problems and the class of fixed-parameter tractable counting problems; the intended meaning will always be clear from the context.

4.1 #W$[1]$-complete problems

Theorem 14 *The following problems are complete for* #W$[1]$ *under parameterized parsimonious reductions:*

(1) #WSAT(2-CNF),

(2) p-#CLIQUE, p-#SUB, p-#HOM, p-#EMB,

(3) p-#$(\Pi_0[\tau])$ *for every vocabulary* τ *that is not monadic.*

(4) p-#HALT *("Count the* k*-step accepting computation paths of a nondeterministic Turing machine").*

Proof: Basically, the proof of these results amounts to checking that the many-one reductions proving the W$[1]$-completeness of the corresponding decision problems are parsimonious (or can be made parsimonious by simple modifications). Some of these reductions are quite simple, and we can sketch them here. For those that are more complicated, we just give appropriate references.

A *conjunctive query* is a first-order formula of the form $\exists x_1 \ldots \exists x_k(\alpha_1 \wedge \ldots \wedge \alpha_\ell)$, where $\alpha_1, \ldots, \alpha_\ell$ are atoms. In particular, a *quantifier free* conjunctive query is just a conjunction of atoms. We denote the class of all conjunctive queries by CQ and the class of all quantifier free conjunctive queries by Π_0-CQ. If Φ is a class of formulas, then Φ[binary] is the class of all formulas $\varphi \in \Phi$ whose vocabulary is at most binary.

We will first establish the following chain of reductions:

$$p\text{-\#}(\Pi_0\text{-CQ}) \leq^{\text{fp}}_{\text{pars}} p\text{-\#Hom} \leq^{\text{fp}}_{\text{pars}} p\text{-\#Emb} \leq^{\text{fp}}_{\text{pars}} p\text{-\#}(\Pi_0\text{-CQ}) \qquad (\star)$$

p-#$(\Pi_0$-CQ$) \leq^{\text{fp}}_{\text{pars}} p$-#HOM :
With every formula $\varphi(x_1, \ldots, x_k) \in \Pi_0$-CQ of vocabulary τ we associate a $\tau \cup \{\text{EQ}\}$-structure $\mathcal{A}_\varphi$, where EQ is a binary relation symbol not contained in τ. The universe of $\mathcal{A}_\varphi$ is $\{x_1, \ldots, x_k\}$, the set of variables of φ. For $R \in \tau$, say, r-ary, $R^{\mathcal{A}_\varphi}$ is the set of all tuples $(x_{i_1}, \ldots, x_{i_r})$ such that $Rx_{i_1} \ldots x_{i_r}$ is an atom of φ. Moreover, EQ$^{\mathcal{A}_\varphi}$ is the set of all pairs (x_{i_1}, x_{i_2}) such that $x_{i_1} = x_{i_2}$ is an atom of φ. Note that $||\mathcal{A}_\varphi|| \in O(|\varphi|)$.

For a τ-structure $\mathcal{B}$ we let $\mathcal{B}_{\text{EQ}}$ be the $\tau \cup \{\text{EQ}\}$-expansion of $\mathcal{B}$ in which EQ is interpreted by the equality relation on B. Then it is easy to see that for all $(b_1, \ldots, b_k) \in B^k$ we have $(b_1, \ldots, b_k) \in \varphi(\mathcal{B})$ if, and only if, the mapping $x_i \mapsto b_i$, for $1 \leq i \leq k$, is a homomorphism from $\mathcal{A}_\varphi$ to $\mathcal{B}$. This yields a parsimonious reduction from p-#$(\Pi_0$-CQ$)$ to p-#HOM.

p-#HOM $\leq^{\text{fp}}_{\text{pars}}$ p-#EMB :
Suppose we have structures $\mathcal{A}$ and $\mathcal{B}$ and want to count the homomorphisms from $\mathcal{A}$ to $\mathcal{B}$. Let τ be the vocabulary of $\mathcal{A}$ and $\mathcal{B}$ and $\tau^* = \tau \cup \{P_a \mid a \in A\}$, where for every $a \in A$, P_a is a new unary relation symbol that is not contained in τ. Let $\mathcal{A}^*$ be the τ^*-expansion of $\mathcal{A}$ with $P_a = \{a\}$ for $a \in A$. We can view $\mathcal{A}^*$ as the expansion of $\mathcal{A}$ where each element gets its individual colour. We let $\mathcal{B}^*$ the be following τ^*-structure: The universe of $\mathcal{B}^*$ is $A \times B$. For r-ary $R \in \tau$ and $(a_1, b_1), \dots, (a_r, b_r) \in A \times B$ we let $\big((a_1, b_1), \dots, (a_r, b_r)\big) \in R^{\mathcal{B}^*}$ if, and only if, $(b_1, \dots, b_r) \in R^{\mathcal{B}}$. For $a \in A$ let $P_a^{\mathcal{B}^*} = \{a\} \times B$. For a homomorphism $h : \mathcal{A} \to \mathcal{B}$ we let $h^* : A^* \to B^*$ be the mapping defined by $h^*(a) = (a, h(a))$. It is easy to see that the mapping $h \mapsto h^*$ is a bijection between the homomorphisms from $\mathcal{A}$ to $\mathcal{B}$ and the embeddings from $\mathcal{A}^*$ to $\mathcal{B}^*$.

p-#EMB $\leq^{\text{fp}}_{\text{pars}}$ p-#(Π_0-CQ) :
For every τ-structure $\mathcal{A}$ we define a formula $\varphi_{\mathcal{A}} \in \Pi_0$-CQ of vocabulary $\tau \cup \{\text{NEQ}\}$, where NEQ is a new binary relation symbol. Suppose that $\mathcal{A} = \{a_1, \dots, a_k\}$. The formula $\varphi_{\mathcal{A}}$ has variables $x_1, \dots, x_k$. For every r-ary $R \in \tau$ and every tuple $(a_{i_1}, \dots, a_{i_r}) \in R^{\mathcal{A}}$, $\varphi_{\mathcal{A}}$ contains the atom $Rx_{i_1} \dots x_{i_r}$. In addition, $\varphi_{\mathcal{A}}$ contains the atoms NEQ$x_i x_j$ for $1 \leq i < j \leq k$.

For a τ-structure $\mathcal{B}$ we let $\mathcal{B}_{\text{NEQ}}$ be the $\tau \cup \{\text{NEQ}\}$-expansion of $\mathcal{B}$ in which NEQ is interpreted by the inequality relation on B. Then it is easy to see that for all mappings $h : A \to B$ we have: h is an embedding of $\mathcal{A}$ into $\mathcal{B}$ if, and only if, $\big(h(a_1), \dots, h(a_k)\big) \in \varphi_{\mathcal{A}}(\mathcal{B})$.

Next, we establish the following chain of reductions (for every $d \geq 1$):

$$\begin{aligned} &\#\text{WSAT}(\Gamma_{1,d}) \leq^{\text{fp}}_{\text{pars}} p\text{-}\#(\Pi_0\text{-CQ}) \leq^{\text{fp}}_{\text{pars}} p\text{-}\#(\Pi_0\text{-CQ[binary]}) \\ &\leq^{\text{fp}}_{\text{pars}} p\text{-}\#\text{CLIQUE} \leq^{\text{fp}}_{\text{pars}} \#\text{WSAT(2-CNF)} \leq^{\text{fp}}_{\text{pars}} \#\text{WSAT}(\Gamma_{1,1}) \end{aligned} \tag{$\star\star$}$$

Together with ($\star$), this proves the #W[1]-completeness of all problems listed in (1) and (2) except p-#SUB.

#WSAT($\Gamma_{1,d}$) $\leq^{\text{fp}}_{\text{pars}}$ p-#(Π_0-CQ):
The proof of Lemma 21 in [19] showing that WSAT($\Gamma_{1,d}$) is fixed-parameter many-one reducible to p-MC(CQ) yields a parsimonious reduction from #WSAT($\Gamma_{1,d}$) to p-#(Π_0-CQ).

p-#(Π_0-CQ) $\leq^{\text{fp}}_{\text{pars}}$ p-#(Π_0-CQ[binary]) :
The proof of Lemma 17 in [19] showing that p-MC(CQ) is fixed-parameter many-one reducible to p-MC(CQ[binary]) yields the claimed parsimonious reduction. Here and in later proofs we use (variants of) the following observation: Let $\varphi(\bar{x})$ and $\psi(\bar{x}, \bar{y})$ be formulas and $\mathcal{A}$ a structure. If for all tuples $\bar{a} \in A$ we have

$$\mathcal{A} \models \varphi(\bar{a}) \quad \Longleftrightarrow \quad \mathcal{A} \models \exists \bar{y} \psi(\bar{a}, \bar{y}),$$

and for all tuples $\bar{a} \in A$ there exists at most one tuple $\bar{b} \in A$ such that $\mathcal{A} \models \psi(\bar{a}, \bar{b})$, then $|\varphi(\mathcal{A})| = |\psi(\mathcal{A})|$.

p-#(Π_0-CQ[binary]) $\leq^{\text{fp}}_{\text{pars}}$ p-#CLIQUE : The reduction in Proposition 22 of [19] is parsimonious.

p-#CLIQUE $\leq^{\text{fp}}_{\text{pars}}$ #WSAT(2-CNF) :
Let $\mathcal{G}$ be a graph. For every $a \in G$ let X_a be a propositional variable. Set

$$\alpha_{\mathcal{G}} \quad = \bigwedge_{a,b \in G, a \neq b, (a,b) \notin E^{\mathcal{G}}} (\neg X_a \vee \neg X_b) \ \wedge \ \bigwedge_{a \in G} (X_a \vee \neg X_a).$$

Then $\alpha_{\mathcal{G}}$ is (equivalent to) a formula in 2-CNF. The second part of the formula ensures that every variable X_a with $a \in G$ occurs in $\alpha_{\mathcal{G}}$. The number of cliques of size k is just the number of assignments of weight k satisfying $\alpha_{\mathcal{G}}$.

#WSAT(2-CNF) $\leq^{\text{fp}}_{\text{pars}}$ #WSAT($\Gamma_{1,1}$) :
2-CNF is a subset of $\Gamma_{1,1}$, so the reduction is trivial.

This completes the proof of ($\star\star$). We next show (3). Let τ be a vocabulary that is not monadic. We leave it to the reader to show that p-#CLIQUE $\leq^{\text{fp}}_{\text{pars}}$ p-#($\Pi_0[\tau]$).

p-#$(\Pi_0[\tau]) \leq^{\text{fp}}_{\text{pars}}$ p-#$(\Pi_0$-CQ) :
Let $\mathcal{A}$ be a τ-structure and $\varphi \in \Pi_0[\tau]$. We can assume that $\varphi = \varphi_1 \vee \ldots \vee \varphi_m$, where each φ_i is a conjunction of literals and $\varphi_i \wedge \varphi_j$ is unsatisfiable for all $i \neq j$. Let $\tau' := \tau \cup \{\bar{R} \mid R \in \tau\} \cup \{\text{EQ}, \text{NEQ}\}$, where for all $R \in \tau$ the symbol $\bar{R}$ is a new relation symbol of the same arity as R and EQ, NEQ are new binary relation symbols. Let $\mathcal{A}'$ be the τ'-expansion of $\mathcal{A}$ in which $\bar{R}$ is interpreted as the complement of $R^{\mathcal{A}}$ and EQ and NEQ are interpreted as equality and inequality, respectively. Since the vocabulary is fixed, $\mathcal{A}'$ can be computed from $\mathcal{A}$ in polynomial time. If n is the size of the universe of $\mathcal{A}$, then computing the relations $\text{EQ}^{\mathcal{A}'}$ and $\text{NEQ}^{\mathcal{A}'}$ requires quadratic time, and for an r-ary $R \in \tau$, computing $\bar{R}^{\mathcal{A}'} = A^r \setminus R^{\mathcal{A}}$ requires time $O(n^r)$.

Let φ' be the formula obtained by replacing positive literals of the form $x = y$ by $\text{EQ}xy$ and by replacing negative literals by positive ones in the obvious way using the new relation symbols $\bar{R}$ and NEQ. Then $\varphi' = \varphi'_1 \vee \ldots \vee \varphi'_m$, where each φ'_i is a conjunction of atoms (i.e., positive literals). Note that $\varphi(\mathcal{A}) = \varphi'(\mathcal{A}')$ and

$$\text{for } \bar{a} \in \mathcal{A}' \text{ there is at most one } i \text{ with } \mathcal{A}' \models \varphi'_i(\bar{a}). \qquad (\star\star\star)$$

Finally we want to get rid of the disjunctions in φ'. For this purpose we introduce a structure $\mathcal{A}''$ essentially consisting of m copies of $\mathcal{A}'$, the ith one taking care of φ'_i. More precisely: let $\tau'' := \{\hat{R} \mid R \in \tau'\} \cup \{<, T\}$, where $\text{arity}(\hat{R}) = \text{arity}(R) + 1$ and where $<$ and T are binary. Define the τ''-structure $\mathcal{A}''$ by

$$\begin{aligned}
A'' &:= \{1, \ldots, m\} \cup (\{1, \ldots, m\} \times A) \\
<^{\mathcal{A}''} &:= \text{the natural ordering on } \{1, \ldots, m\} \\
T^{\mathcal{A}''} &:= \{((i, a), (i, b)) \mid 1 \leq i \leq m,\ a, b \in A\} \\
\hat{R}^{\mathcal{A}''} &:= \{(i, (i, a_1), \ldots, (i, a_{\text{arity}(R)})) \mid 1 \leq i \leq m,\ R^{\mathcal{A}'} a_1 \ldots a_{\text{arity}(R)}\} \cup \\
& \ \{(i, (j, a_1), \ldots, (j, a_{\text{arity}(R)})) \mid 1 \leq i, j \leq m,\ i \neq j,\ a_1, \ldots, a_{\text{arity}(R)} \in A\}.
\end{aligned}$$

Moreover set

$$\varphi''(x_1, \ldots, x_k, y_1, \ldots, y_m) \quad := \quad y_1 < \ldots < y_m \wedge \bigwedge_{1 \leq \ell \leq \ell' \leq k} T x_\ell x_{\ell'} \wedge \bigwedge_{i=1}^{m} \varphi'_i \frac{\hat{R} y_i \bar{z}}{R \bar{z}},$$

where $\varphi'_i \frac{\hat{R} y_i \bar{z}}{R \bar{z}}$ is obtained from φ'_i by replacing, for all $R \in \tau'$, atomic subformulas of the form $R\bar{z}$ by $\hat{R} y_i \bar{z}$. Clearly, $\varphi(\bar{x}, \bar{y}) \in \Pi_0$-CQ. By $(\star\star\star)$, we have $|\varphi(\mathcal{A})| = |\varphi''(\mathcal{A}'')|$.

Next, we prove the #W[1]-completeness of p-#SUB. We observe that the number of substructures of a structure $\mathcal{B}$ that are isomorphic to a structure $\mathcal{A}$ equals the number of embeddings of $\mathcal{A}$ into $\mathcal{B}$ divided by the number of automorphisms of $\mathcal{A}$. Unfortunately, this does not immediately yield a parsimonious reduction from p-#SUB to p-#EMB or vice versa. However, p-#CLIQUE is a restriction of p-#SUB, thus we have p-#CLIQUE $\leq^{\text{fp}}_{\text{pars}}$ p-#SUB.

To prove that p-#SUB is in #W[1], we reduce p-#SUB to p-#EMB. Let $\mathcal{A}, \mathcal{B}$ be τ-structures and let $<$ be a binary relation symbol not contained in τ. Let us call a $\tau \cup \{<\}$-structure $\mathcal{C}$ in which $<^{\mathcal{C}}$ is a linear order of the universe an *ordered* $\tau \cup \{<\}$*-structure*. Let $\mathcal{A}_1, \ldots, \mathcal{A}_m$ be a list of expansions of $\mathcal{A}$ to ordered $\tau \cup \{<\}$-structures such that

(i) for $1 \leq i < j \leq m$, the structures $\mathcal{A}_i$ and $\mathcal{A}_j$ are not isomorphic,

(ii) every expansion $\mathcal{A}'$ of $\mathcal{A}$ to an ordered $\tau \cup \{<\}$-structure is isomorphic to an $\mathcal{A}_i$ for some $i, 1 \leq i \leq m$.

Thus $\mathcal{A}_1, \ldots, \mathcal{A}_m$ is a list of all ordered expansions of $\mathcal{A}$, where each isomorphism type is only listed once.

Let $\mathcal{B}_<$ be an arbitrary expansion of $\mathcal{B}$ to an ordered $\tau \cup \{<\}$-structure. Then

$$|\{\mathcal{A}' \subseteq \mathcal{B} \mid \mathcal{A}' \cong \mathcal{A}\}| = \sum_{i=1}^{m} |\{\mathcal{A}' \subseteq \mathcal{B}_< \mid \mathcal{A}' \cong \mathcal{A}_i\}|.$$

Moreover, for each i the number of substructures of $\mathcal{B}_<$ isomorphic to $\mathcal{A}_i$ is equal to the number of embeddings of $\mathcal{A}_i$ into $\mathcal{B}_<$.

Let $\prec$ be another binary relation symbol not contained in $\tau \cup \{<\}$ and $\tau^* = \tau \cup \{<, \prec\}$. Let $\mathcal{A}^*$ be the τ^*-structure obtained by taking the disjoint union of $\mathcal{A}_1, \dots, \mathcal{A}_m$ and defining $\prec^{\mathcal{A}}$ such that for all $a_i \in A_i, a_j \in A_j$ we have $a_i \prec^{\mathcal{A}^*} a_j$ if, and only if, $i < j$. For $1 \leq i \leq m$, let $\mathcal{B}_i^*$ be the τ^*-structure obtained by replacing the copy of $\mathcal{A}_i$ in $\mathcal{A}^*$ by a copy of $\mathcal{B}_<$. Then the number of embeddings of $\mathcal{A}^*$ into $\mathcal{B}_i^*$ is equal to the number of embeddings of $\mathcal{A}_i$ into $\mathcal{B}_<$. Finally let $\mathcal{B}^*$ be the disjoint union of $\mathcal{B}_1^*, \dots, \mathcal{B}_m^*$. Then the number of embeddings of $\mathcal{A}^*$ into $\mathcal{B}^*$ is equal to the sum of the numbers of embeddings of $\mathcal{A}^*$ into $\mathcal{B}_i^*$ for $1 \leq i \leq m$. Putting everything together, the number of substructures of $\mathcal{B}$ isomorphic to $\mathcal{A}$ is equal to the number of embeddings of $\mathcal{A}^*$ into $\mathcal{B}^*$.

It remains to prove #W[1]-completeness of p-#HALT. The proof of Theorem 8.3 in [15] implicitly contains parsimonious reductions from p-#(Π_0-CQ[binary]) to p-#HALT and from p-#HALT to p-#(Π_0-CQ). □

The decision versions of all problems mentioned in Theorem 14 are W[1]-complete under parameterized many-one reductions. The following theorem is interesting because it is not known whether the decision problem p-MC(Π_0) is contained in the closure of W[1] under parameterized T-reductions.

Theorem 15 *p-#(Π_0) is contained in the closure of #W[1] under parameterized T-reductions.*

Proof: We shall prove that $p\text{-}\#(\Pi_0) \leq_{\mathrm{T}}^{\mathrm{fp}} p\text{-}\#(\Pi_0\text{-CQ})$.

Note that the reduction from p-#($\Pi_0[\tau]$) to p-#(Π_0-CQ) we gave in the proof of Theorem 14 does not yield a parameterized parsimonious reduction from p-#(Π_0) to p-#(Π_0-CQ), because if the vocabulary is not fixed in advance the structure $\mathcal{A}'$ can get much larger than $\mathcal{A}$.

At least, the same argument as given in the proof of Theorem 14 shows that we can restrict our attention to conjunctions of literals (instead of arbitrary quantifier free formulas). Consider a formula

$$\varphi = \alpha_1 \wedge \ldots \wedge \alpha_\ell \wedge \neg\beta_1 \wedge \ldots \wedge \neg\beta_m,$$

where $\alpha_1, \dots, \alpha_\ell, \beta_1, \dots, \beta_m$ are atoms. The crucial observation is that for any structure $\mathcal{A}$ we have

$$\begin{aligned} |\varphi(\mathcal{A})| \quad = \quad & \big|(\alpha_1 \wedge \ldots \wedge \alpha_\ell \wedge \neg\beta_1 \wedge \ldots \wedge \neg\beta_{m-1})(\mathcal{A})\big| \\ & -\big|(\alpha_1 \wedge \ldots \wedge \alpha_\ell \wedge \neg\beta_1 \wedge \ldots \wedge \neg\beta_{m-1} \wedge \beta_m)(\mathcal{A})\big| \end{aligned}$$

Note that the two formulas on the left hand side of the equality have fewer negated atoms than φ. We can now recursively reduce the number of negated atoms in these two formulas using the same trick until we end up with a family of quantifier free conjunctive queries. This gives us a parameterized Turing reduction from p-#(Π_0) to p-#(Π_0-CQ). □

4.2 A machine characterisation of #W[1] As it is also the case for many other parameterized complexity classes, the definition of the classes #W[t] is a bit unsatisfactory because all the classes are only defined as the closure of a certain problem under a certain type of reduction. In particular, one may ask why we chose parsimonious reductions and not, say, Turing reductions. Indeed, McCartin [23] defined her version of the classes #W[t] using a different form of reductions, and that makes the theory seem a bit arbitrary. Compare this with the situation for the class #P, which has a natural machine characterisation: A classical counting problem $F : \Sigma^* \to \mathbb{N}$ is in #P if, and only if, there is a polynomial time non-deterministic Turing machine N such that for every instance x of the problem, $F(x)$ is the number of accepting paths of N on input x.

Recently, a machine characterisation of the class W[1] was given [8]. In this subsection, we adapt this characterisation to give a characterisation of #W[1] along the lines of the above mentioned characterisation of #P.

The machine model we use, which has been introduced in [8], is based on the standard random access machines (RAMs) described in [25]. The arithmetic operations are addition, subtraction, and division by two (rounded off), and we use a uniform cost measure. The model is non-standard when it comes to nondeterminism. A *nondeterministic RAM* is a RAM with an additional instruction "GUESS i j" whose semantics is: Guess a natural number less than or equal to the number stored in register i and store it in register j. Acceptance of an input by a nondeterministic RAM program is defined as usually for nondeterministic machines. Steps of a computation of a nondeterministic RAM that execute a GUESS instruction are called *nondeterministic steps*.

Following [8], we call a nondeterministic RAM program $\mathbb{P}$ a *W-program*, if there is a computable function f and a polynomial p such that for every input (x, k) with $|x| = n$ the program $\mathbb{P}$ on every run

(1) performs at most $f(k) \cdot p(n)$ steps;

(2) at most $f(k)$ steps are nondeterministic;

(3) at most the first $f(k) \cdot p(n)$ registers are used;

(4) at every point of the computation the registers contain numbers $\leq f(k) \cdot p(n)$.

We call a W-program $\mathbb{P}$ a *W*$[1]$*-program*, if there is a computable function h such that for every input (x, k), for every run of $\mathbb{P}$

(5) all nondeterministic steps are among the last $h(k)$ steps.

Theorem 16 (Chen, Flum, and Grohe [8]) *Let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized decision problem. Then $Q \in \mathrm{W}[1]$ if, and only if, there is $\mathrm{W}[1]$-program deciding Q.*

The main result of this section is a counting version of this theorem:

Theorem 17 *Let $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$ be a parameterized counting problem. Then $F \in \#\mathrm{W}[1]$ if, and only if, there is a $\mathrm{W}[1]$-program $\mathbb{P}$ such that, for all $(x, k) \in \Sigma^* \times \mathbb{N}$, $F(x, k)$ is the number of accepting paths of $\mathbb{P}$ on input (x, k).*

Proof: First assume that $F \in \#\mathrm{W}[1]$. Then, by Theorem 14, there is a parsimonious reduction from F to p-#HALT. Hence, there are computable functions f, g, a polynomial p, and an algorithm assigning to every instance (x, k) of F, in time $\leq f(k) \cdot p(n)$, a nondeterministic Turing machine $M = M_{x,k}$ and a natural number $k' = k'(x, k) \leq g(k)$ such that $F(x, k)$ is the number of accepting paths of M of length k'.

We can assume that the states and the symbols of the alphabet of M are natural numbers $\leq f(k) \cdot p(n)$. We define a W-program $\mathbb{P}$ that on input $(x, k) \in \Sigma^* \times \mathbb{N}$ proceeds as follows:

1. It computes M and k';

2. It guesses a sequence of k' configurations of M;

3. It verifies that the sequence of guessed configurations forms an accepting computation of M.

We can do this, in particular line 1, with a W-program using our parameterized parsimonious reduction from F to p-#HALT. Moreover, the number of steps needed by line 2 and line 3 is bounded by $h(k)$ for a suitable computable function h. Finally, the number of accepting paths of $\mathbb{P}$ is exactly the number of accepting paths of M.

Assume now that we have a $\mathrm{W}[1]$-program $\mathbb{P}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$, $F(x, k)$ is the number of accepting paths of $\mathbb{P}$ on input (x, k). Let f, p, h witness that $\mathbb{P}$ is a $\mathrm{W}[1]$-program. For every instance $(x, k) \in \Sigma^* \times \mathbb{N}$ of F we shall define a non-deterministic Turing machine $M = M_{x,k}$ and an integer k' such that $F(x, k)$ is the number of accepting paths of M of length at most k'. Of course we have to do this in such a way that the mapping $(x, k) \mapsto (M, k')$ is a parameterized reduction.

So let $(x, k) \in \Sigma^* \times \mathbb{N}$ and $n = |x|$. The alphabet of $M = M_{x,k}$ contains $0, 1, \ldots, f(k) \cdot p(n)$. Thus alphabet symbols can be used to represent register content and register addresses of all runs of $\mathbb{P}$ on input (x, k). In addition, the alphabet contains a few control symbols. The transition function of M will be defined in such a way that M simulates the computation of $\mathbb{P}$ on input (x, k) *from the first non-deterministic step onwards*. The content of all the registers before the first non-deterministic step is hardwired into M. The changes of the register contents during the at most $h(k)$ non-deterministic steps are written on the worktape, so eventually the worktape contains pairs $(i_1, a_1), \ldots, (i_\ell, a_\ell)$ in any order, where (i_j, a_j) indicates that the current content of register i_j is a_j, and $\ell \leq h(k)$. For more details on the definition of M we refer the reader to [8]. $\square$

4.3 #W$[1]$ and counting satisfying assignments of a 3-CNF-formula The following theorem gives further evidence that #W$[1] \neq$ FPT, because it seems unlikely that counting the satisfying assignments of a 3-CNF-formula with n variables is possible in time $2^{o(n)}$. A decision version of this theorem has been proved by Abrahamson, Downey and Fellows [1].

Theorem 18 *If* #W$[1] =$ FPT *then there is an algorithm counting the satisfying assignments of a* 3-CNF-*formula with* n *variables in time* $2^{o(n)}$.

Proof: Suppose that #W$[1] =$ FPT. Then #WSAT(3-CNF) is in FPT. Thus there is an algorithm solving #WSAT(3-CNF) in time $f(k) \cdot n^c$ for some computable function $f : \mathbb{N} \to \mathbb{N}$ and constant c. Then there exists a function $g : \mathbb{N} \to \mathbb{N}$ such that

(i) $f(g(n)) \leq 2^{o(n)}$,

(ii) $\lim_{n\to\infty} g(n) = \infty$,

(iii) $g(n)$ can be computed in time $2^{o(n)}$.

Let $\gamma = \bigwedge_{i=1}^{m} \delta_i$, where each clause δ_i is a disjunction of at most 3 literals, be a formula in 3-CNF, and let $\mathcal{X} = \{X_1, \ldots, X_n\}$ be the set of variables of γ. We assume that no clause appears twice; thus we have $m \leq (2n)^3$. We want to compute the number of satisfying assignments of γ in time $2^{o(n)}$.

Let $k = g(n)$. Note that (ii) implies $n/k \leq o(n)$; we will use this repeatedly in the following argument. For $1 \leq j \leq k$, let

$$\mathcal{X}_j = \Big\{ X_i \;\Big|\; (j-1) \cdot \frac{n}{k} < i \leq j \cdot \frac{n}{k} \Big\}.$$

For every $S \subseteq \mathcal{X}_j$, let Y_j^S be a new variable. Let $\mathcal{Y}_j$ be the set of all Y_j^S and $\mathcal{Y} = \bigcup_{j=1}^{k} \mathcal{Y}_j$. Then

$$|\mathcal{Y}| \leq k \cdot 2^{\lceil n/k \rceil} \leq 2^{o(n)}.$$

Call a truth value assignment to the variables in $\mathcal{Y}$ *good* if for $1 \leq j \leq k$ exactly one variable in $\mathcal{Y}_j$ is set to true. There is a bijection I between the truth value assignments to the variables in $\mathcal{X}$ and the good truth value assignments to the variables in $\mathcal{Y}$ defined by

$$I(A)(Y_j^S) = \text{TRUE} \iff \forall X \in \mathcal{X}_j : \big(A(X) = \text{TRUE} \iff X \in S \big),$$

for all $A : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$, $1 \leq j \leq k$, and $S \subseteq \mathcal{X}_j$. Let

$$\beta = \bigwedge_{\substack{1 \leq j \leq k \\ S,T \subseteq \mathcal{X}_j, S \neq T}} (\neg Y_j^S \vee \neg Y_j^T)$$

and note that $|\beta| \leq k \cdot \big(2^{\lceil n/k \rceil}\big)^2 \leq 2^{o(n)}$. Observe that the weight k assignments to the variables in $\mathcal{Y}$ satisfying β are precisely the good assignments. Thus there is a bijection between the weight k satisfying assignments for β and the assignments to the variables in $\mathcal{X}$.

For $1 \leq j \leq k$ and every variable $X \in \mathcal{X}_j$, let

$$\alpha_X = \bigwedge_{S \subseteq \mathcal{X}_j, X \notin S} \neg Y_j^S,$$
$$\alpha_{\neg X} = \bigwedge_{S \subseteq \mathcal{X}_j, X \in S} \neg Y_j^S.$$

and observe that for every assignment $A : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$ we have

$$A(X) = \text{TRUE} \iff I(A) \text{ satisfies } \alpha_X$$
$$\iff I(A) \text{ does not satisfy } \alpha_{\neg X}$$

Let γ' be the formula obtained from γ by replacing each literal X by the formula α_X and each literal $\neg X$ by $\alpha_{\neg X}$. Then for every assignment $A : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$ we have

$$A \text{ satisfies } \gamma \iff I(A) \text{ satisfies } \gamma'.$$

By applying de Morgan's rule to each clause δ_j of γ (or rather to the disjunction of conjunctions δ_j has become in γ') we can turn γ' into an equivalent conjunction of at most

$$m \cdot \left(2^{\lceil n/k \rceil}\right)^3$$

disjunctions of at most 3 literals each. Let γ'' be this 3-CNF-formula and $\gamma^* = \beta \wedge \gamma''$. Then I is a bijection between the satisfying assignments of γ and the weight k satisfying assignments of γ^*.

By our initial assumption, we can compute the number of weight k satisfying assignments of γ^* in time $f(k) \cdot (n^*)^c$, where $n^* = |\mathcal{Y}| \leq 2^{o(n)}$ is the number of variables of γ^*. Since $f(k) = f(g(n)) \leq 2^{o(n)}$, this shows that we can compute the number of satisfying assignments of γ in time $2^{o(n)}$. □

4.4 #W$[2]$-complete problems

Theorem 19 *The following problems are complete for* #W$[2]$ *under parameterized parsimonious reductions:*

(1) #WSAT(CNF)*,*

(2) p-#DOMINATING SET,

(3) p-#$(\Pi_{1,1}[\tau])$ *for every vocabulary* τ *that is not monadic.*

The equivalence between (1) in (3) in Theorem 19 can be lifted to the other classes of the #W-hierarchy, but we only deal with #W[2] here.

Proof (of Theorem 19): Let $\Pi_{1,1}[s]$ denote the class of all formulas in $\Pi_{1,1}$ whose vocabulary is at most s-ary. We will establish the following chain of reductions for every $d \geq 0$ and $s \geq 2$:

$$\text{\#WSAT}(\Gamma_{2,d}) \leq^{\text{fp}}_{\text{pars}} p\text{-\#}(\Pi_{1,1}[s]) \leq^{\text{fp}}_{\text{pars}} p\text{-\#DOMINATING SET} \leq^{\text{fp}}_{\text{pars}} \text{\#WSAT(CNF)}.$$

Recalling that CNF $\subseteq \Gamma_{2,0}$ and observing that p-#DOMINATING SET $\leq^{\text{fp}}_{\text{pars}}$ p-#$(\Pi_{1,1}[\tau])$ for every vocabulary τ that is not monadic, we see that this proves the theorem.

#WSAT$(\Gamma_{2,d}) \leq^{\text{fp}}_{\text{pars}} p$-#$(\Pi_{1,1}[2])$:
By standard means one can show that there is a d' only depending on d such that every formula in $\Gamma_{2,d}$ is equivalent to a formula of the form

$$\alpha = \bigwedge_{i \in I} \delta_i,$$

where for some $d_\alpha \leq d'$ every δ_i is a disjunction of conjunctions of exactly d_α literals,

$$\delta_i = \bigvee_{j \in J^i} \beta_{ij}$$

with

$$\beta_{ij} = \lambda_{ij1} \wedge \ldots \wedge \lambda_{ijd_\alpha}. \qquad (\star)$$

So let such an α, say with variables $X_1, \ldots, X_n$, and a $k \in \mathbb{N}$ be given. If we have an assignment of weight k setting $X_{i_1}, \ldots, X_{i_k}$ with $i_1 < \ldots < i_k$ TRUE and satisfying β_{ij} as in $(\star)$, then the positive literals in β_{ij} must be among $X_{i_1}, \ldots, X_{i_k}$. Thus for every negative literal $\neg X_r$ in β_{ij} we must have $r < i_1$ or $i_k < r$ or $i_s < r < i_{s+1}$ for some s. We use this fact in our reduction appropriately.

For $m \in \mathbb{N}$ set $[m] := \{1, \ldots, m\}$ and

$$[m]_2 \quad := \quad \{(i,j) \mid 0 \leq i < j \leq m+1\}.$$

For a set M and $m \in \mathbb{N}$ denote by $\text{Pow}_{\leq m}(M) := \{Y \subseteq M \mid |Y| \leq m\}$ the set of subsets of M of cardinality $\leq m$.

We let $\tau = \{<, \prec, E, \text{FIRST}, \text{LAST}, F, \text{DISJ}, \text{SAT}\}$ with binary $<, \prec, E, F, \text{SAT}$ and unary FIRST, LAST, DISJ. Let $\mathcal{A}_\alpha$ be the following τ-structure: The universe is

$$A_\alpha \quad := \quad [n] \cup [n]_2 \cup \text{Pow}_{\leq d_\alpha}([n] \cup [n]_2) \cup \{\delta_i \mid i \in I\}.$$

Recall that I is the index set of the conjunction in the formula α. The selection of $i \in [n]$ means that the variable X_i gets the value TRUE and the selection of $(i,j) \in [n]_2$ means that all variables X_ℓ with $i < \ell < j$ get the truth value FALSE.

The relations of $\mathcal{A}_\alpha$ are specified by:

$$\begin{array}{rcl}
<^{\mathcal{A}_\alpha} & := & \text{the natural ordering on } [n] \\
\prec^{\mathcal{A}_\alpha} & := & \text{a total ("lexicographic") ordering on } \text{Pow}_{\leq d_\alpha}([n] \cup [n]_2); \\
E^{\mathcal{A}_\alpha} & := & \{(j,(i,j)) \mid 0 \leq i < j \leq n+1\} \cup \{(i,(i,j)) \mid 0 \leq i < j \leq n+1\}; \\
\text{FIRST}^{\mathcal{A}_\alpha} & := & \{(0,j) \mid 0 \leq j \leq n+1\}; \\
\text{LAST}^{\mathcal{A}_\alpha} & := & \{(i,n+1) \mid 0 \leq i \leq n+1\}; \\
F^{\mathcal{A}_\alpha} & := & \{(i,M) \mid i \in [n],\, M \in \text{Pow}_{\leq d_\alpha}([n] \cup [n]_2),\, i \in M\} \cup \\
 & & \{((i,j),M) \mid (i,j) \in [n]_2,\, M \in \text{Pow}_{\leq d_\alpha}([n] \cup [n]_2),\, (i,j) \in M\}; \\
\text{DISJ}^{\mathcal{A}_\alpha} & := & \{\delta_i \mid i \in I\}; \\
\text{SAT}^{\mathcal{A}_\alpha} & := & \{(M,\delta_i) \mid M \in \text{Pow}_{\leq d_\alpha}([n] \cup [n]_2),\, i \in I, \text{there is a } j \in J^i \text{ such that for } s = 1,\dots,n\text{:} \\
 & & \quad \text{if } X_s \text{ is a literal of } \beta_{ij} \text{ then } s \in M \text{ and} \\
 & & \quad \text{if } \neg X_s \text{ is a literal of } \beta_{ij} \text{ then there is } (\ell,m) \in M \text{with } \ell < s < m\}.
\end{array}$$

Let $r := |\text{Pow}_{\leq d_\alpha}([2 \cdot k + 1])|$. Note that $\|\mathcal{A}_\alpha\| \leq \|\mathcal{A}\|^c$, where $c = c(d)$, and $r \leq g(d,k)$ for some computable function g.

The number of satisfying assignments of α of weight k is $|\varphi_{\alpha,k}(\mathcal{A}_\alpha)|$, where $\varphi_{\alpha,k}(x_1,\dots,x_k,z_1,\dots,z_{k+1},u_1,\dots,u_r)$ is the $\Pi_{1,1}$-formula

$$\begin{aligned}
\varphi_{\alpha,k} = \forall y \Big(& x_1 < \ldots < x_k \wedge \textstyle\bigwedge_{i=1}^{k} (E x_i z_i \wedge E x_i z_{i+1}) \wedge \text{FIRST}\, z_1 \wedge \text{LAST}\, z_{k+1} \\
& \wedge u_1 \prec \ldots \prec u_r \wedge \textstyle\bigwedge_{i=1}^{r} \big(F y u_i \rightarrow (\bigvee_{j=1}^{k} y = x_j \vee \bigvee_{j=1}^{k+1} y = z_j) \big) \\
& \wedge (\text{DISJ}\, y \rightarrow \textstyle\bigvee_{j=1}^{r} \text{SAT} u_j y) \Big).
\end{aligned}$$

p-#$(\Pi_{1,1}[s]) \leq^{\text{fp}}_{\text{pars}}$ p-#DOMINATING SET :
For notational simplicity, we assume $s = 2$. Let τ be a vocabulary that only contains unary and binary relation symbols. Assume we are given a τ-structure $\mathcal{A}$ with universe A and a $\Pi_{1,1}[\tau]$-formula

$$\varphi(x_1,\dots,x_\ell) \quad = \quad \forall y \psi(x_1,\dots,x_\ell,y).$$

Let $\mathcal{G} = (G, E^{\mathcal{G}})$ be the graph defined as follows: The vertex set is

$$G := (\{1,\dots,\ell\} \times A) \,\dot\cup\, A^\ell \,\dot\cup\, (A \times \{0\}) \,\dot\cup\, \{b_i^j \mid 1 \leq i \leq \ell, 1 \leq j \leq \ell+2\} \,\dot\cup\, \{b^j \mid 1 \leq j \leq \ell+2\}$$

($\dot\cup$ denotes disjoint union), where b_i^j and b^j are new elements. The edge relation $E^{\mathcal{G}}$ is defined in such a way that

(i) every $(a_1,\dots,a_\ell) \in A^\ell$ is connected to all elements of $\{i\} \times (A \setminus \{a_i\})$ for $1 \leq i \leq \ell$;

(ii) for $\bar a \in A^\ell$ and $(b,0) \in A \times \{0\}$: $\{\bar a, (b,0)\} \in E^{\mathcal{G}} \iff \mathcal{A} \models \psi(\bar a, b)$.

(iii) b_i^j is connected to (i,a) for $1 \leq i \leq \ell$, $1 \leq j \leq \ell+2$, $a \in A$.

(iv) b^j is connected to all $\bar a \in A^\ell$ for $1 \leq j \leq \ell+2$.

We claim that:

- Every dominating set of $\mathcal{G}$ of cardinality $\ell+1$ contains exactly one element of each $\{i\} \times A$, and if we label these elements say (i, a_i) for $1 \le i \le \ell$, then the $(\ell+1)$st element in the dominating set is $(a_1, \dots, a_\ell) \in A^\ell$.
- For all $a_1, \dots, a_\ell \in A$,

$$\big\{(i, a_i) \;\big|\; 1 \le i \le \ell\big\} \cup \{\bar{a}\} \text{ is a dominating set of } \mathcal{G} \iff \mathcal{A} \models \varphi(\bar{a}).$$

To see this, suppose that D is a dominating set of $\mathcal{G}$ of size $\ell+1$. Then D must contain at least one vertex of $\{i\} \times A$ for $1 \le i \le \ell$ and one vertex of A^ℓ, because this is the only way the vertices b_i^j and b^j, for $1 \le j \le \ell+2$ can be dominated with $\ell+1$ vertices. Suppose that D contains the vertices $(1, a_1), \dots, (\ell, a_\ell)$. Let d be the remaining element of D. If $d = (d_1, \dots, d_\ell) \ne (a_1, \dots, a_\ell)$, say $d_1 \ne a_1$, then $(1, d_1)$ is not dominated by $a_1, \dots, a_\ell, d$. Therefore, d must be $(a_1, \dots, a_\ell)$. However, d must also dominate $A \times \{0\}$, and this is only possible if $\mathcal{A} \models \psi(a_1, \dots, a_\ell, b)$ for all $b \in A$.

Thus $|\varphi(\mathcal{A})|$ is the number of dominating sets of $\mathcal{G}$ of cardinality $\ell+1$. But note that $\mathcal{G}$ is too big for a parameterized reduction, since G contains the set A^ℓ, where the exponent depends on the parameter φ. So we need a more refined reduction. We can assume that $\psi(\bar{x}, y) = \psi_1 \wedge \ldots \wedge \psi_m$ where each ψ_i is a disjunction of literals. Each literal contains at most 2 variables. Therefore, we do not need A^ℓ but a copy A_{ij} of A^2 for $1 \le i < j \le \ell$. We replace A^ℓ above by all these copies and $A \times \{0\}$ by $A \times \{1, \dots, m\}$. We replace (i) by

(i') Every (a_i, a_j) in the copy A_{ij} of A^2 is connected to all elements of $\{i\} \times (A \setminus \{a_i\})$ and all elements of $\{j\} \times (A \setminus \{a_j\})$.

Furthermore, we replace (ii) by

(ii') for $1 \le i < j \le \ell$, (a_i, a_j) in the copy A_{ij} of A^2, and for $(b, k) \in A \times \{1, \dots, m\}$:

$$\{(a_i, a_j), (b, k)\} \in E^{\mathcal{G}} \iff \text{there is a literal } \lambda(x_i, x_j, y) \text{ in } \psi_k \text{ whose (at most two) free variables are among } x_i, x_j, y \text{ such that } \mathcal{A} \models \lambda(a_i, a_j, b).$$

Moreover, instead of the b_i^j and the b^j we add for every $i = 1, \dots, \ell$ and for every copy of A^2 a set of $\ell + \binom{\ell}{2} + 1$ new elements that ensure that every dominating set of cardinality $\ell + \binom{\ell}{2}$ contains exactly one element of every $\{i\} \times A$ and of every copy of A^2.

Then dominating sets of cardinality $\ell + \binom{\ell}{2}$ and tuples in $\mathcal{A}$ satisfying φ are related in an one-to-one fashion.

p-#DOMINATING SET $\le^{\text{fp}}_{\text{pars}}$ #WSAT(CNF) :
Let $\mathcal{G} = (G, E^{\mathcal{G}})$ be a graph. For $a \in G$ let X_a be a propositional variable. Let $\alpha_{\mathcal{G}}$ be the propositional formula

$$\alpha_{\mathcal{G}} \;:=\; \bigwedge_{a \in G} \big(X_a \vee \bigvee_{(a,b) \in E} X_b\big).$$

Then, $\alpha_{\mathcal{G}}$ is (equivalent to) a formula in $\Gamma_{2,0}$. Clearly the number of satisfying assignments of $\alpha_{\mathcal{G}}$ of weight k equals the number of dominating sets of $\mathcal{G}$ of size k. □

Remark 20 As opposed to the proof of Theorem 14, the reductions given in the proof of Theorem 19 are not just variants of the standard reductions showing the W[2]-completeness of the respective problems under many-one reductions. As a matter of fact, our proof yields a new proof of the complicated result that p-DOMINATING SET is W[2]-complete under parameterized many-one reductions.

5 Counting cycles and paths

Theorem 21 *The following problems are* #W[1]*-complete under parameterized Turing reductions:*

(1) p-#CYCLE *and* p-#DIRCYCLE *("Count the cycles of length* k *in a (directed) graph").*

(2) p-#PATH *and* p-#DIRPATH *("Count the paths of length* k *in a (directed) graph").*

To be precise, let us define a *path of length* k in a directed graph $(G, E^{\mathcal{G}})$ to be a substructure of G isomorphic to $\big(\{1,\dots,k\}, \{(i,i+1) \mid 1 \le i < k\}\big)$. A *cycle of length* k is a substructure isomorphic to $\big(\{1,\dots,k\}, \{(i,i+1) \mid 1 \le i < k\} \cup \{(k,1)\}\big)$. Paths and cycles in undirected graphs are defined similarly.

Thus all problems in Theorem 21 are restrictions of the substructure problem p-#SUB and thus in #W[1] by Theorem 14. The decision versions of the problems are fixed-parameter tractable. This is an immediate consequence of Plehn and Voigt's [26] theorem that the parameterized embedding problem restricted to graphs of bounded tree-width is fixed-parameter tractable and the fact that paths have tree-width 1 and cycles have tree-width 2.

Lemma 22

$$p\text{-\#DIRCYCLE} \le^{\text{fp}}_{\text{pars}} p\text{-\#CYCLE} \le^{\text{fp}}_{\text{T}} p\text{-\#PATH} \le^{\text{fp}}_{\text{T}} p\text{-\#DIRPATH}.$$

Proof: p-#DIRCYCLE $\le^{\text{fp}}_{\text{pars}}$ p-#CYCLE :
For a directed graph $\mathcal{G}$, let $\mathcal{G}^u_{p,q}$ be the undirected graph obtained from $\mathcal{G}$ by the following two steps:

(1) Replace each vertex a of $\mathcal{G}$ by an undirected path of length p such that the (directed) edges with head a in $\mathcal{G}$ get the first vertex of this path as their new head and the edges with tail a in $\mathcal{G}$ get the last vertex of this path as their new tail.

(2) Replace each directed edge in this graph (corresponding to an edge of $\mathcal{G}$) by an undirected path of length q.

Figure 1 gives an example.

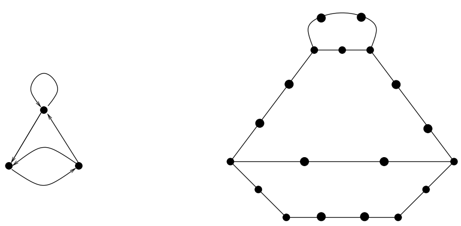

Figure 1. A directed graph $\mathcal{G}$ and the corresponding $\mathcal{G}^u_{2,3}$

Observe that each cycle in $\mathcal{G}^u_{p,q}$ has length $\ell \cdot p + m \cdot q$ for some integers $\ell, m \ge 0$ with $\ell \le m$. Further observe that each directed cycle of length k in $\mathcal{G}$ lifts to a cycle of length $k(p+q)$ in $\mathcal{G}^u_{p,q}$. Given k, we want to choose p and q in such a way that each cycle of length $k(p+q)$ in $\mathcal{G}^u_{p,q}$ is the lifting of a directed cycle of length k in $\mathcal{G}$. To achieve this, we have to choose p and q in such a way that

$$k(p+q) \ne \ell \cdot p + m \cdot q \tag{1}$$

for all $\ell, m \ge 0$ with $\ell < m$. If we choose $p \le q$, then (1) holds for $m > 2k$. So we have to fulfil (1) for $0 \le \ell < m \le 2k$. Hence, we have to avoid $\binom{2k+1}{2}$ linear equalities. Clearly we can find natural numbers $p \le q$ satisfying none of these equalities.

For such p and q, the number of directed cycles of length k in $\mathcal{G}$ equals the number of undirected cycles of length $k(p+q)$ in $\mathcal{G}^u_{p,q}$.

p-#CYCLE $\leq^{\text{fp}}_{\text{T}}$ p-#PATH :
Let $\mathcal{G}$ be an undirected graph and $k \geq 1$. Without loss of generality we can assume that $k \geq 3$ because counting loops in a graph is easy.

For each $e = \{v, w\} \in E^{\mathcal{G}}$ and all $\ell, m \geq 0$, we let $\mathcal{G}_e(\ell, m)$ be the graph obtained from $\mathcal{G}$ by adding vertices as $v_1, \ldots, v_\ell, w_1, \ldots, w_m$ and edges between v_i and w for $1 \leq i \leq \ell$ and between w_j and v for $1 \leq j \leq m$.

We observe that the number x_e of paths of length $k+1$ from v_1 to w_1 in $\mathcal{G}_e(\ell, m)$ is exactly the number of cycles of length k in $\mathcal{G}$ containing the edge e. We now show how to compute x_e from the numbers of paths of length $k+1$ in the graphs $\mathcal{G}_e(\ell, m)$ for $0 \leq \ell, m \leq 1$. This yields a parameterized Turing reduction from p-#CYCLE to p-#PATH.

We observe that the v_i and w_j can only be endpoints of paths in $\mathcal{G}_e(\ell, m)$, and that each path can have at most one endpoint among $v_1, \ldots, v_\ell$ and at most one endpoint among $w_1, \ldots, w_m$ (because each path ending in v_i must go through w and each path ending in w_j must go through v).

We let

- $x = x_e$ be the number of paths of length $(k+1)$ from v_1 to w_1 in $\mathcal{G}_e(1,1)$,
- y be the number of paths of length $(k+1)$ in $\mathcal{G}_e(1,1)$ that contain v_1, but not w_1,
- z be the number of paths of length $(k+1)$ in $\mathcal{G}_e(1,1)$ that contain w_1, but not v_1,
- w be the number of paths of length $(k+1)$ in $\mathcal{G}_e(1,1)$ that neither contain v_1 nor w_1.

Let $p_{\ell m}$ be the number of paths of length $(k+1)$ in $\mathcal{G}_e(\ell, m)$. Then we have

$$p_{\ell m} = w + \ell \cdot m \cdot x + \ell \cdot y + m \cdot z.$$

For $0 \leq \ell, m \leq 1$ we obtain a system of 4 linear equations in the variables w, x, y, z whose matrix is nonsingular. Thus it has a unique solution which, in particular, gives us the desired value x.

p-#PATH $\leq^{\text{fp}}_{\text{T}}$ p-#DIRPATH :
This is trivial; just replace each edge of an undirected graph that is not a loop by two directed edges. Then each path (of length at least 2) in the undirected graph corresponds to exactly two paths of the same length in the directed graph. □

Next, we will prove that p-#CLIQUE $\leq^{\text{fp}}_{\text{T}}$ p-#DIRCYCLE. This requires a sequence of lemmas. Let $h : \mathcal{H} \to \mathcal{G}$ be a homomorphism and, for $i \geq 1$, let k_i be the number of vertices $b \in G$ such that $|h^{-1}(b)| \geq i$. Then $\sum_{i \geq 1} k_i = |H|$. The *type* of h is the polynomial

$$t_h(X) = \prod_{i \geq 1} (X - i + 1)^{k_i} = \prod_{b \in G} (X)_{|h^{-1}(b)|},$$

where the notation $(X)_i$ is used for the "falling factorial", that is, $(X)_0 = 1$ and $(X)_{i+1} = (X)_i(X - i)$ for all $i \geq 0$. In particular, an embedding from $\mathcal{H}$ into $\mathcal{G}$ is a homomorphism of type $X^{|H|}$.

Let $\mathcal{D}_k$ denote the directed cycle of length k whose vertices are $1, \ldots, k$ in cyclic order. We consider the following generalisation of p-#DIRCYCLE:

p-#TDC	
Input:	Directed graph $\mathcal{G}$, polynomial $t(X)$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the homomorphisms $h : \mathcal{D}_k \to \mathcal{G}$ of type $t(X)$.

Lemma 23

$$p\text{-\#TDC} \leq^{\text{fp}}_{\text{T}} p\text{-\#DIRCYCLE}$$

Proof: For a directed graph $\mathcal{G}$ and natural numbers $\ell, m \geq 1$, let $\mathcal{G}_{\ell,m}$ be the graph obtained from $\mathcal{G}$ as follows:

- The universe of $\mathcal{G}_{\ell,m}$ is
$$G_{\ell,m} = G \times \{1, \ldots, \ell\} \times \{1, \ldots, m\}.$$
- There is an edge from (a, i, j) to (a', i', j') in $\mathcal{G}_{\ell,m}$ either if $i = \ell$ and $i' = 1$ and there is an edge from a to a' in $\mathcal{G}$ or if $a = a'$ and $i' = i + 1$.

Figure 2 gives an example.

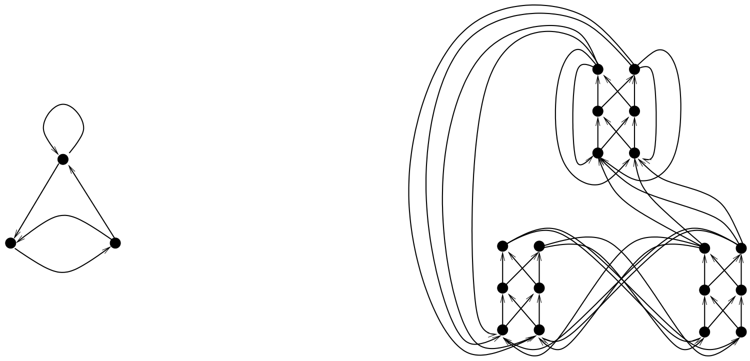

Figure 2. A directed graph $\mathcal{G}$ and the corresponding $\mathcal{G}_{3,2}$

Recall that the vertices of the cycle $\mathcal{D}_k$ are $1, \ldots, k$. The *projection* of an embedding $e : \mathcal{D}_{k\cdot\ell} \to \mathcal{G}_{\ell,m}$ is the homomorphism $\pi(e) : \mathcal{D}_k \to \mathcal{G}$ which maps vertex $a \in D_k$ to the first component of $e((a-1)\cdot\ell+1)$, that is, we let $\pi(e)(a) = b$ if $e((a-1)\cdot\ell+1) = (b, i, j)$ for some $i \in \{1, \ldots, \ell\}, j \in \{1, \ldots, m\}$.

Observe that for every homomorphism $h : \mathcal{D}_k \to \mathcal{G}$ there are

$$\ell \cdot t_h(m)^\ell$$

embeddings $e : \mathcal{D}_{k\cdot\ell} \to \mathcal{G}_{\ell,m}$ with projection $\pi(e) = h$. Let T be the set of all types of homomorphisms from $\mathcal{D}_k$ into some graph. For every type $t \in T$, let x_t be the number of homomorphisms $h : \mathcal{D}_k \to \mathcal{G}$ with $t_h = t$. Then

$$b_\ell = \sum_{t\in T} x_t \cdot \ell \cdot t(m)^\ell$$

is the number of embeddings $e : \mathcal{D}_{k\cdot\ell} \to \mathcal{G}_{\ell,m}$.

The types in T are polynomials of degree at most k. Thus for distinct $t(X), t'(X) \in T$ there are at most k distinct $x \in \mathbb{N}$ such that $t(x) = t'(x)$. Therefore, there is an $m \leq k \cdot |T|^2$ such that for all distinct $t(X), t'(X) \in T$ we have $t(m) \neq t'(m)$. We fix such an m.

We let $\vec{b} = (b_1, \ldots, b_{|T|})$, $\vec{x} = (x_t)_{t\in T}$, and $A = (a_{\ell t})_{\substack{1\leq\ell\leq|T|\\ t\in T}}$, where $a_{\ell t} = \ell \cdot t(m)^\ell$. Then

$$A \cdot \vec{x} = \vec{b}.$$

Since the matrix $(\frac{1}{\ell} a_{\ell t})_{\substack{1\leq\ell\leq|T|\\ t\in T}}$ is a Vandermonde matrix and thus nonsingular, the matrix A is also nonsingular, and thus

$$\vec{x} = A^{-1}\vec{b}.$$

Now our Turing reduction from p-#TDC to p-#DIRCYCLE works as follows:

1 Compute the set T and a suitable m.

2 For $1 \leq \ell \leq |T|$, compute the graph $\mathcal{G}_{\ell,m}$.

3 For $1 \leq \ell \leq |T|$, compute the number b_ℓ of embeddings $e : \mathcal{D}_{k\cdot\ell} \to \mathcal{G}_{\ell\cdot m}$ (using the oracle to p-#DIRCYCLE and noting that b_ℓ is $k \cdot \ell$ times the number of cycles of length $k \cdot \ell$ in $\mathcal{G}_{\ell\cdot m}$).

4 Compute the matrix A and solve the system $A \cdot \vec{x} = \vec{b}$.

5 Return x_t, where $t(X)$ is the input polynomial. (If $t \notin T$, then return 0.)

Since the set T, the number m, and the matrix A only depend on the parameter k, this is a parameterized Turing reduction. □

For $k, \ell \geq 1$, let $\Omega(k, \ell)$ denote the space of all mappings $f : \{1, \ldots, k \cdot \ell\} \to \{1, \ldots, k\}$ such that $|f^{-1}(i)| = \ell$ for $1 \leq i \leq k$.

Lemma 24 *Let $k \geq 1$, and let $\mathcal{H} = (H, E^{\mathcal{H}})$ be a directed graph with universe $H = \{1, \ldots, k\}$ and $E^{\mathcal{H}} \neq H^2$. Then*

$$\lim_{\ell\to\infty} \Pr_{f\in\Omega(k,\ell)} \big(f \text{ is a homomorphism from } \mathcal{D}_{k\cdot\ell} \text{ to } \mathcal{H}\big) = 0$$

(where f is chosen uniformly at random).

Proof: Let $(x, y) \in H^2 \setminus E^{\mathcal{H}}$ and $m \leq k \cdot \ell$. We call a tuple $(i_1, \ldots, i_m) \in H^m$ *good* if $(i_j, i_{j+1}) \neq (x, y)$ for $1 \leq j \leq m - 1$ and *bad* otherwise. For $(i_1, \ldots, i_m) \in H^m$ chosen uniformly at random we have

$$\Pr\big((i_1, \ldots, i_m) \text{ good}\big) \leq \Pr\big(\forall j, 1 \leq j \leq m/2 : \ (i_{2j-1}, i_{2j}) \neq (x, y)\big) = \left(1 - \frac{1}{k^2}\right)^{\lfloor m/2 \rfloor}.$$

Furthermore, for all $i_1, \ldots, i_m \in H$ we have

$$\Pr_{f\in\Omega(k,\ell)} \big(\forall j, 1 \leq j \leq m : \ f(j) = i_j\big) \leq \left(\frac{\ell}{k\cdot\ell - m}\right)^m.$$

To see this inequality, note that a choosing a random function $f \in \Omega(k, \ell)$ can be modeled by randomly picking kl balls without repetitions out of a bin that intially contains l balls each of colours $1, \ldots, k$. The probability that the ith ball ist of color j at most

$$\frac{\ell}{k \cdot \ell - (i - 1)},$$

because at most ℓ of the remaining $k \cdot \ell - (i - 1)$ are of colour j. Now the inequality follows straightforwardly.

Thus

$$\begin{aligned}
&\Pr_{f\in\Omega(k,\ell)}(f \text{ is a homomorphism from } \mathcal{D}_{k\cdot\ell} \text{ to } \mathcal{H})\\
\leq &\sum_{(i_1,\ldots,i_m)\in H^m \text{ good}} \Pr_{f\in\Omega(k,\ell)}(f(j) = i_j \text{ for } 1 \leq j \leq m)\\
\leq &\sum_{(i_1,\ldots,i_m)\in H^m \text{ good}} \left(\frac{\ell}{k\cdot\ell - m}\right)^m\\
= &\left(\frac{\ell}{k\cdot\ell - m}\right)^m \cdot k^m \cdot \Pr_{(i_1,\ldots,i_m)\in H^m}\big((i_1, \ldots, i_m) \text{ good}\big)\\
\leq &\left(\frac{k\cdot\ell}{k\cdot\ell - m}\right)^m \cdot \left(1 - \frac{1}{k^2}\right)^{\lfloor m/2 \rfloor}.
\end{aligned}$$

Let $\varepsilon > 0$. Then there exists an $m(\varepsilon, k)$ such that for $m \geq m(\varepsilon, k)$ we have

$$\left(1 - \frac{1}{k^2}\right)^{\lfloor m/2 \rfloor} \leq \frac{\varepsilon}{2}.$$

Moreover, for every m there exists an $\ell(m)$ such that for $\ell \geq \ell(m)$ we have

$$\left(\frac{k \cdot \ell}{k \cdot \ell - m}\right)^m = \left(\frac{1}{1 - \frac{m}{k \cdot \ell}}\right)^m \leq \frac{1}{\left(1 - \frac{m}{\ell}\right)^m} \leq 2.$$

Thus for all $\ell \geq \ell(m(\varepsilon, k))$ we have

$$\Pr_{f \in \Omega(k,\ell)}(f \text{ is a homomorphism from } \mathcal{D}_{k \cdot \ell} \text{ to } \mathcal{H}) \leq \varepsilon.$$

□

Lemma 25

$$p\text{-\#C}{\small\text{LIQUE}} \leq^{\text{fp}}_{\text{T}} p\text{-\#TDC}.$$

Proof: Let $k \geq 1$. For a graph $\mathcal{H}$, let $\overset{\leftrightarrow}{\mathcal{H}}$ denote the directed graph with the same vertex set and edge set

$$\big\{(a, a) \bigm| a \in H\big\} \cup \big\{(a, b) \bigm| \{a, b\} \in E^{\mathcal{H}}\big\}.$$

For every graph $\mathcal{H}$ with k vertices and every $\ell \geq 1$, let $a_{\mathcal{H}\ell}$ be the number of homomorphisms of type $(X)^k_\ell$ from $\mathcal{D}_{k \cdot \ell}$ into $\overset{\leftrightarrow}{\mathcal{H}}$ (that is, homomorphisms for which each point in the image has exactly ℓ pre-images). Let $\overset{\mathbb{N}}{a}_{\mathcal{H}} = (a_{\mathcal{H}1}, a_{\mathcal{H}2}, \ldots)$ and, for every $\ell \geq 1$, $\overset{\ell}{a}_{\mathcal{H}} = (a_{\mathcal{H}1}, a_{\mathcal{H}2} \ldots, a_{\mathcal{H}\ell})$. We consider $\overset{\mathbb{N}}{a}_{\mathcal{H}}$ and $\overset{\ell}{a}_{\mathcal{H}}$ as vectors in the vector spaces $\mathbb{Q}^{\mathbb{N}}$ and $\mathbb{Q}^{\ell}$, respectively.

Let $k \geq 1$, and let $\mathcal{K}$ be the complete graph with vertices $\{1, \ldots, k\}$. Let $\mathbb{H}$ be the set of all graphs with vertex set $\{1, \ldots, k\}$, where up to isomorphism each graph occurs only once in $\mathbb{H}$, and let $\mathbb{H}^- = \mathbb{H} \setminus \{\mathcal{K}\}$.

For a set S of vectors in $\mathbb{Q}^{\mathbb{N}}$ or $\mathbb{Q}^{\ell}$, we let $\langle S \rangle$ denote the linear span of S.

Claim 1:

$$\overset{\mathbb{N}}{a}_{\mathcal{K}} \notin \Big\langle \big\{ \overset{\mathbb{N}}{a}_{\mathcal{H}} \bigm| \mathcal{H} \in \mathbb{H}^- \big\} \Big\rangle.$$

Proof: Recall that $\Omega(k, \ell)$ denotes the set of all mappings $h : \{1, \ldots, k \cdot \ell\} \to \{1, \ldots, k\}$ with the property that $|h^{-1}(i)| = \ell$ for $1 \leq i \leq k$.

We first observe that for all $\ell \geq 1$,

$$a_{\mathcal{K}\ell} = |\Omega(k, \ell)|.$$

On the other hand, by Lemma 24 for all graphs $\mathcal{H} \in \mathbb{H}^-$ we have

$$\lim_{\ell \to \infty} \frac{a_{\mathcal{H}\ell}}{|\Omega(k, \ell)|} = 0.$$

Suppose for contradiction that

$$\overset{\mathbb{N}}{a}_{\mathcal{K}} = \sum_{i=1}^{n} \lambda_i \overset{\mathbb{N}}{a}_{\mathcal{H}_i}$$

for graphs $\mathcal{H}_1, \ldots, \mathcal{H}_n \in \mathbb{H}^-$. Choose ℓ sufficiently large such that for $1 \leq i \leq n$

$$\frac{a_{\mathcal{H}_i\ell}}{a_{\mathcal{K}\ell}} = \frac{a_{\mathcal{H}_i\ell}}{|\Omega(k, \ell)|} < \frac{1}{\sum_{i=1}^{n} |\lambda_i|}.$$

Then

$$a_{\mathcal{K}\ell} = \sum_{i=1}^{n} \lambda_i a_{\mathcal{H}_i\ell} \leq a_{\mathcal{K}\ell} \sum_{i=1}^{n} |\lambda_i| \frac{a_{\mathcal{H}_i\ell}}{a_{\mathcal{K}\ell}} < a_{\mathcal{K}\ell} \sum_{i=1}^{n} |\lambda_i| \frac{1}{\sum_{j=1}^{n} |\lambda_j|} = a_{\mathcal{K}\ell},$$

which is a contradiction. This proves Claim 1.

Claim 2: There is an $\ell = \ell(k) \in \mathbb{N}$ such that

$$\overset{\ell}{a}_{\mathcal{K}} \notin \left\langle \left\{ \overset{\ell}{a}_{\mathcal{H}} \mid \mathcal{H} \in \mathbb{H}^- \right\} \right\rangle.$$

Furthermore, the mapping $k \mapsto \ell(k)$ is computable.

Proof: For $\iota \in \mathbb{N} \cup \{\mathbb{N}\}$, let

$$V_\iota = \left\langle \left\{ \overset{\iota}{a}_{\mathcal{H}} \mid \mathcal{H} \in \mathbb{H}^- \right\} \right\rangle.$$

Identifying $(a_1, \ldots, a_i) \in \mathbb{Q}^i$ with $(a_1, \ldots, a_i, 0, 0, \ldots) \in \mathbb{Q}^{\mathbb{N}}$, for all $i \geq 1$ we can view V_i a subspace of V_j for all $j \geq i$ and of $V_{\mathbb{N}}$. Thus we can find an increasing sequence

$$\mathbb{B}_1 \subseteq \mathbb{B}_2 \subseteq \mathbb{B}_3 \subseteq \cdots \subseteq \mathbb{H}^-$$

such that for all $i \geq 1$,

$$\left\{ \overset{i}{a}_{\mathcal{H}} \;\middle|\; \mathcal{H} \in \mathbb{B}_i \right\}$$

is a basis of V_i. Since $V_{\mathbb{N}}$ is a finite dimensional vector space, there is an $n \in \mathbb{N}$ such that $\mathbb{B}_i = \mathbb{B}_n$ for all $i \geq n$.

Now if $\overset{i}{a}_{\mathcal{K}}$ was in V_i for all $i \geq 1$, then for all $i \geq 1$, the vector $\overset{i}{a}_{\mathcal{K}}$ could be written as a unique linear combination of the vectors in $\left\{ \overset{i}{a}_{\mathcal{H}} \;\middle|\; \mathcal{H} \in \mathbb{B}_i \right\}$. For all $i \geq n$, these linear combinations would be identical, thus $\overset{\mathbb{N}}{a}_{\mathcal{K}}$ would be in $V_{\mathbb{N}}$. This contradicts Claim 1 and thus proves that for some $\ell \in \mathbb{N}$,

$$\overset{\ell}{a}_{\mathcal{K}} \notin V_\ell.$$

Clearly such an ℓ is computable from k, since we can compute all vectors $\overset{i}{a}_{\mathcal{H}}$ for $\mathcal{H} \in \mathbb{H}$ and $i \in \mathbb{N}$. This completes the proof of Claim 2.

Now we are ready to prove the lemma. Let $k \geq 1$ and define $\mathcal{K}$, $\mathbb{H}$, $\mathbb{H}^-$, and the vectors $\overset{\iota}{a}_{\mathcal{H}}$ as above. Choose $\ell = \ell(k)$ according to Claim 2.

Let $\mathcal{G}$ be a graph. For every graph $\mathcal{H} \in \mathbb{H}$, let $x_{\mathcal{H}}$ be the number of subsets $A \subseteq G$ such that the subgraph induced by $\mathcal{G}$ on A is isomorphic to $\mathcal{H}$. We want to determine the number $x_{\mathcal{K}}$. For $1 \leq i \leq \ell$, let b_i be the number of homomorphisms from $\mathcal{D}_{k \cdot i}$ into $\overset{\leftrightarrow}{\mathcal{G}}$ of type $(X)_i^k$, and let $\overset{\ell}{b} = (b_1, \ldots, b_\ell)$. The numbers b_i can be computed by an oracle to p-#TDC.

Observe that for $1 \leq i \leq \ell$ we have

$$b_i = \sum_{\mathcal{H} \in \mathbb{H}} x_{\mathcal{H}} a_{\mathcal{H}i}$$

and thus

$$\overset{\ell}{b} = \sum_{\mathcal{H} \in \mathbb{H}} x_{\mathcal{H}} \overset{\ell}{a}_{\mathcal{H}}.$$

Since $\overset{\ell}{a}_{\mathcal{K}}$ is linearly independent from $\left\{ \overset{\ell}{a}_{\mathcal{H}} \;\middle|\; \mathcal{H} \in \mathbb{H}^- \right\}$, the coefficient $x_{\mathcal{K}}$ can be computed by solving this system of linear equations. □

Proof of Theorem 21: The theorem follows immediately from Lemmas 22, 23, 25 and Theorem 14. □

6 Conclusions

We have set up a framework for a parameterized complexity theory of counting problems and proved a number of completeness results. In particular, we proved the fixed-parameter intractability of natural counting problems whose decision version is fixed-parameter tractable.

A lot of interesting problems remain open, let us just mention two of them:

- In view of Valiant's #P-completeness result for counting perfect matchings, it would be quite nice to show that the parameterized matching problem p-#MATCHING is #W$[1]$-complete. We conjecture that this is the case.
- Another interesting question is related to Toda's theorem: Does #W$[1]$ contain the whole W-hierarchy, or maybe even the A-hierarchy (introduced in [15])?

References

[1] K.A. Abrahamson, R.G. Downey, and M.R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogs. *Annals of pure and applied logic*, 73:235–276, 1995.

[2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42:844–856, 1995.

[3] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.

[4] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.

[5] V. Arvind and V. Raman. Approximation algorithms for some parameterized counting problems. In P. Bose and P. Morin, editors, *Proceedings of the 13th Annual International Symposium on Algorithms and Computation*, volume 2518 of *Lecture Notes in Computer Science*, pages 453–464. Springer-Verlag, 2002.

[6] H.L. Bodlaender, R.G. Downey, M.R. Fellows, M.T. Hallett, and H.T. Wareham. Parameterized complexity analysis in computational biology. *Computer Applications in the Biosciences*, 11:49–57, 1995.

[7] L. Cai and J. Chen. On fixed-parameter tractability and approximability of NP optimization problems. *Journal of Computer and System Sciences*, 54:465–474, 1997.

[8] Y. Chen, J. Flum, and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 13–29, 2003.

[9] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 194–242. Elsevier Science Publishers, 1990.

[10] B. Courcelle, J.A. Makowsky, and U. Rotics. On the fixed-parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1–2):23–52, 2001.

[11] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995.

[12] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141:109–131, 1995.

[13] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[14] R.G. Downey, M.R. Fellows, and K. Regan. Descriptive complexity and the W-hierarchy. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetic*, volume 39 of *AMS-DIMACS Volume Series*, pages 119–134. AMS, 1998.

[15] J. Flum and M. Grohe. Fixed-parameter tractability, definability, and model checking. *SIAM Journal on Computing*, 31(1):113–145, 2001.

[16] M. Frick. Generalized model-checking over locally tree-decomposable classes. In H. Alt and A. Ferreira, editors, *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2285 of *Lecture Notes in Computer Science*, pages 632–644. Springer-Verlag, 2002.

[17] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, 48:1184–1206, 2001.

[18] G. Gottlob, N. Leone, and M. Sideri. Fixed-parameter complexity in AI and nonmonotonic reasoning. In M. Gelfond, N. Leone, and G. Pfeifer, editors, *Logic Programming and Nonmonotonic Reasoning, 5th International Conference, LPNMR'99*, volume 1730 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1999.

[19] M. Grohe. The parameterized complexity of database queries. In *Proceedings of the 20th ACM Symposium on Principles of Database Systems*, pages 82–92, 2001.

[20] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 712–721, 2001.

[21] Ph.G. Kolaitis and M.N. Thakur. Approximation properties of NP minimization classes. *Journal of Computer and System Sciences*, 50:391–411, 1995.

[22] J.A. Makowsky. Colored Tutte polynomials and Kauffman brackets for graphs of bounded tree width. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 487–495, 2001.

[23] C. McCartin. Parameterized counting problems. In K. Diks and W. Rytter, editors, *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, volume 2420 of *Lecture Notes in Computer Science*, pages 556–567. Springer-Verlag, 2002.

[24] C. H. Papadimitriou and M. Yannakakis. On the complexity of database queries. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 12–19, 1997.

[25] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[26] J. Plehn and B. Voigt. Finding minimally weighted subgraphs. In R. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, WG '90*, volume 484 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, 1990.

[27] U. Stege. *Resolving Conflicts in Problems from Computational Biology*. PhD thesis, ETH Zuerich, 2000. PhD Thesis No.13364.

[28] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[29] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

Appendix A: A list of problems appearing in this paper

Vertex cover and related problems A *vertex cover* of a graph $\mathcal{G} = (G, E^{\mathcal{G}})$ is a subset $X \subseteq G$ such that for all edges $(v, w) \in E^{\mathcal{G}}$ either $v \in X$ or $w \in X$.

p-VERTEX COVER
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if $\mathcal{G}$ has a vertex cover of size k.

p-#VERTEX COVER
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Count the vertex covers of $\mathcal{G}$ of size k.

A *dominating set* of a graph $\mathcal{G} = (G, E^{\mathcal{G}})$ is a subset $X \subseteq G$ such that for all vertices $w \in G$ either $w \in X$ or $(v, w) \in E^{\mathcal{G}}$ for some $v \in X$.

p-DOMINATING SET
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if $\mathcal{G}$ has a dominating set of size k.

p-#DOMINATING SET
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Count the dominating sets of $\mathcal{G}$ of size k.

In general, the *standard parameterization* of an optimisation problem is the parameterized decision problem asking whether there exists a solution of size k, where k is the quantity to be optimised and the parameter. The counting version can be defined accordingly.

Homomorphisms, embeddings, and substructures

p-HOM
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Decide if there exists a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

p-#HOM
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Count the homomorphisms from $\mathcal{A}$ to $\mathcal{B}$.

p-EMB
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Decide if there exists an embedding of $\mathcal{A}$ into $\mathcal{B}$.

p-#EMB
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Count the embeddings of $\mathcal{A}$ into $\mathcal{B}$.

p-SUB
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Decide if $\mathcal{B}$ has a substructure isomorphic to $\mathcal{A}$.

p-#SUB
Input: Structures $\mathcal{A}$ and $\mathcal{B}$.
Parameter: $||\mathcal{A}||$.
Problem: Count the substructures of $\mathcal{B}$ isomorphic to $\mathcal{A}$.

A *clique* in a graph $\mathcal{G}$ is a subset X of G such that for all distinct $v, w \in X$, $(v, w) \in E^{\mathcal{G}}$.

p-CLIQUE
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if $\mathcal{G}$ has a clique of size k.

p-#CLIQUE
Input: Graph $\mathcal{G}$.
Parameter: $k \in \mathbb{N}$.
Problem: Count the cliques of $\mathcal{G}$ of size k.

p-Path	
Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ contains a path of length k.

p-#Path	
Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the paths of length k in $\mathcal{G}$.

p-DirPath	
Input:	Directed graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ contains a directed path of length k.

p-#DirPath	
Input:	Directed graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the directed paths of length k in $\mathcal{G}$.

p-Cycle	
Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ contains a cycle of length k.

p-#Cycle	
Input:	Graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the cycles of length k in $\mathcal{G}$.

p-DirCycle	
Input:	Directed graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ contains a directed cycle of length k.

p-#DirCycle	
Input:	Directed graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the directed cycles of length k in $\mathcal{G}$.

A *matching* of a graph is a set of edges that pairwise have no endpoint in common.

p-Matching	
Input:	Bipartite graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if $\mathcal{G}$ contains a matching of size k.

p-#Matching	
Input:	Bipartite graph $\mathcal{G}$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the matchings of size k in $\mathcal{G}$.

Logically defined problems The *weight* of an assignment S for the variables of a propositional formula is the number of variables set to TRUE by S. Let Θ be a class of propositional formulas

WSat(Θ)	
Input:	$\theta \in \Theta$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Decide if θ has a satisfying assignment of weight k.

#WSat(Θ)	
Input:	$\theta \in \Theta$.
Parameter:	$k \in \mathbb{N}$.
Problem:	Count the satisfying assignments of θ of weight k.

$|\varphi|$ denotes the length of a formula φ. Let Φ be a class of first-order formulas.

p-MC(Φ)	
Input:	Structure $\mathcal{A}$, formula $\varphi \in \Phi$.
Parameter:	$|\varphi|$.
Problem:	Decide if $\varphi(\mathcal{A}) \neq \emptyset$.

p-#(Φ)	
Input:	Structure $\mathcal{A}$, formula $\varphi \in \Phi$.
Parameter:	$|\varphi|$.
Problem:	Compute $|\varphi(\mathcal{A})|$.

Let $\varphi(X)$ be a formula of vocabulary $\tau \cup \{X\}$.

p-FD($\varphi(X)$)

Input: τ-structure $\mathcal{A}$.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if $\varphi(\mathcal{A})$ contains a relation of size k.

p-#FD($\varphi(X)$)

Input: Structure $\mathcal{A}$.
Parameter: $k \in \mathbb{N}$.
Problem: Count the number of relations of size k in $\varphi(\mathcal{A})$.

The parameterized halting problem

p-HALT

Input: Nondeterministic Turing machine M.
Parameter: $k \in \mathbb{N}$.
Problem: Decide if M accepts the empty word in at most k steps.

p-#HALT

Input: Nondeterministic Turing machine M.
Parameter: $k \in \mathbb{N}$.
Problem: Count the accepting computation paths of M of length at most k for the empty word.